# Dual-Attention Generative Adversarial Network and Flame and Smoke Analysis

**Yuchuan Li**

Thesis submitted to the University of Ottawa
in partial Fulfillment of the requirements for the
Master of Applied Science in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
University of Ottawa

# Abstract

Flame and smoke image processing and analysis could improve performance to detect smoke or fire and identify many complicated fire hazards, eventually to help firefighters to fight fires safely. Deep Learning applied to image processing has been prevailing in recent years among image-related research fields. Fire safety researchers also brought it into their studies due to its leading performance in image-related tasks and statistical analysis. From the perspective of input data type, traditional fire research is based on simple mathematical regressions or empirical correlations relying on sensor data, such as temperature. However, data from advanced vision devices or sensors can be analyzed by applying deep learning beyond auxiliary methods in data processing and analysis. Deep Learning has a bigger capacity in non-linear problems, especially in high-dimensional spaces, such as flame and smoke image processing. We propose a video-based real-time smoke and flame analysis system with deep learning networks and fire safety knowledge. It takes videos of fire as input and produces analysis and prediction for flashover of fire.

Our system consists of four modules. The Color2IR Conversion module is made by deep neural networks to convert RGB video frames into InfraRed (IR) frames, which could provide important thermal information of fire. Thermal information is critically important for fire hazard detection. For example, 600 °C marks the start of a flashover. As RGB cameras cannot capture thermal information, we propose an image conversion module from RGB to IR images. The core of this conversion is a new network that we innovatively proposed: Dual-Attention Generative Adversarial Network (DAGAN), and it is trained using a pair of RGB and IR images.

Next, Video Semantic Segmentation Module helps extract flame and smoke areas from the scene in the RGB video frames. We innovated to use synthetic RGB video data generated and captured from 3D modeling software for data augmentation.

After that, a Video Prediction Module takes the RGB video frames and IR frames as input and produces predictions of the subsequent frames of their scenes.

Finally, a Fire Knowledge Analysis Module predicts if flashover is coming or not, based on fire knowledge criteria such as thermal information extracted from IR images, temperature increase rate, the flashover occurrence temperature, and increase rate of lowest temperature.

For our contributions and innovations, we introduce a novel network, DAGAN, by applying foreground and background attention mechanisms in the image conversion module to help reduce the hardware device requirement for flashover prediction. Besides, we also make use of combination of thermal information from IR images and segmentation information from RGB images in our system for flame and smoke analysis. We also apply a hybrid design of deep neural networks and a knowledge-based system to achieve high accuracy. Moreover, data augmentation is also applied on the Video Semantic Segmentation Module by introducing synthetic video data for training.

The test results of flashover prediction show that our system has leading places quantitative and qualitative in terms of various metrics compared with other existing approaches. It can give a flashover prediction as early as 51 seconds with 94.5% accuracy before it happens.

# Acknowledgment

Firstly, I would like to express my sincere gratitude to my supervisor, Professor WonSook Lee, for the support and guidance during my master's study. It is my pleasure to join LIII++ as it provides a positive, joyful, and unity environment for my study career as well as daily life. I would also appreciate the opportunities that she offered to collaborate with a research institute off-campus. I would certainly not have made so much progress without her suggestions and help.

I am grateful to Dr. Yoon Ko for her help and support with my work on the thesis. It is my pleasure to have a chance and to join her group in the Fire Safety Unit, Construction Research Centre, National Research Council, Canada, Ottawa as a student. She not only gave me guidance in fire science research but also set an example for me and my future career as an excellent researcher.

I would like to say thanks to all my colleagues in LIII++. I had a great time with you, whether in academic discussions like our weekly ML Seminar or coffee break in our labs. I have learned a lot through the discussion and studies with you.

I am thankful for my parents, as they provide financial and emotional support in my two years of study at Ottawa. I would also appreciate my friends for their help.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1. Introduction

## 1.1 Motivation

Fire is one of the most dangerous disasters for human society. The increasing number of buildings has boosted this problem to a higher dangerous level each year. For many families, a fire may have a catastrophic impact on their subsequent life, including the economic pressure caused by property loss and the mental impact caused by family members' injury and death. Reports [1] show that in a developed country like the United States of America, there are an estimated 1,318,500 fire cases in 2018. For home fires, they have caused 2720 civilian fire deaths and 15,200 civilian fire injuries. The total number of direct property losses due to fires is over $25 billion.

The fire also has a profound adverse effect on the firefighters who have been fighting on the front line of the fire scene all year round. More than 30,000 firefighters are injured each year during firefighting operations [2, 3]. As fire has characteristics of fast spreading, the major challenge in firefighting is to put out the fire at an early or initial stage rather than to spend hours trying to extinguish the fire that has spread in the vicinity. An example of a comparison of different stages of fire is shown in **Figure 1-1**. The risk posed to firefighters and human lives increases as time goes on.

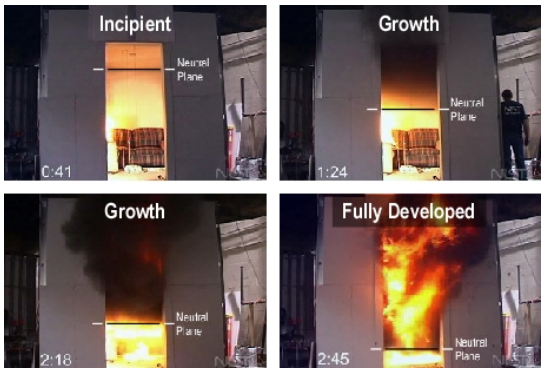


**Figure 1-1:** Examples of different stages of fire, from [4]. At 2:18, the size of the fire was not big yet, but at 2:45, the fire suddenly expanded.

For modern firefighting activities, there are 2 types of bases for data. The first one is built-in sensors in the buildings, such as smoke sensors. The system relies on the fact that smoke caused by the fire would rise up and trigger the smoke sensors installed at the ceilings of the buildings [5]. They are cheap but are only for one-time usage. The second type of data is video data, such as videos captured from cameras taken with firefighters.

Compared with sensor data, video data are more timely and could also provide more information about the fire. For example, in RGB video data, fire can be generally characterized as orange or yellow flames which move from side to side. Smoke could be characterized as a combination of white, grey, and black plumes containing tiny soot particles or burnt particles [6]. Besides, IR videos could provide thermal conditions of a fire, which has far more detail than the temperature from sensor data.

In order to process and analyze the video data of fire, a number of methods have been proposed for fire and smoke processing and detection. One type of is fire safety knowledge-based system. A common hardware requirement for them is high-precision IR cameras. They use feature extraction and segmentation techniques to extract the required features from the RGB or IR images. Then, these features would be further analyzed based on fire safety knowledge for different purposes, such as detection or prediction. Handcrafted features are widely used in smoke or flame detection systems [7, 8, 9, 10, 11]. Besides, the motion between frames is good support for contextual information to analyze and process flame or smoke patterns in images [12, 13, 14, 15, 16]. Furthermore, a fusion of InfraRed (IR) and RGB videos could provide more information on fire [17, 18]. Those methods could achieve high accuracy with powerful IR videos from IR cameras.

However, feasibility also needs to be considered when it comes to firefighting usage. High-precision and powerful IR cameras are usually big and heavy because they need shields and careful protection from severe heat waves. Besides, their price is high, starting from 70 thousand dollars, which are only affordable to big research labs. The cheaper ones are not equipped with high-precision sensors and thus could not capture high temperatures of fire with high accuracy, which could not help firefighters. As knowledge-based models highly depend on the accuracy of IR cameras, IR cameras become limitations of knowledge-based models for firefighters' usage.

On the contrary, Deep Learning could free up equipment requirements and help build a feasible solution for firefighters. Deep Learning is a hot topic in almost all current research areas. It has demonstrated capabilities that surpass traditional and human-level methods in image-related research in recent years, especially in image recognition, segmentation, and classification tasks. Seeing the potential of Deep Learning, researchers for fire science also tried to integrate it into their works with image-related analysis, and it opened up new research ideas. For example, some researchers build systems based on Convolutional Neural Networks (CNNs) for fast and accurate processing of flame or smoke patterns [19, 20, 21, 22]. In addition, some others take advantage of Recurrent Neural Networks (RNNs) in processing sequence data from the past to the future and build models with it [23, 24]. Deep neural networks could also help in the conversion of RGB images to IR images. With deep learning techniques, the equipment requirement could be reduced, and firefighters would only need to bring an RGB camera with them, the thermal information would be generated by deep neural networks, which helps achieve accuracy of the same level with knowledge-based systems. But this time, it is more convenient and affordable.

However, a pure deep learning based system has its own limitation in computational cost. Deep neural networks need to run on powerful Graphic Processing Units (GPUs). They have high power consumption and are not easy to deploy. They will also become slow if we run them on microchips. As a result, a hybrid design system with knowledge-based and deep learning based is the best choice for flame and smoke analysis in firefighting.

Thus, we proposed an RGB Images-based Flame and Smoke Analysis System combining deep neural networks and fire safety knowledge to provide real-time analysis for flame and smoke. It only requires an RGB camera which is feasible and convenient for firefighters' usage and could get the same level of thermal information with conversion by deep neural networks.

## 1.2 Overview of Our System

The input of our system is an RGB video captured from a vision camera. Then the video will be cut into frames for analysis purposes in the system. Our purpose is, analyzing those frames in real-time based on the information of current and previous input frames.

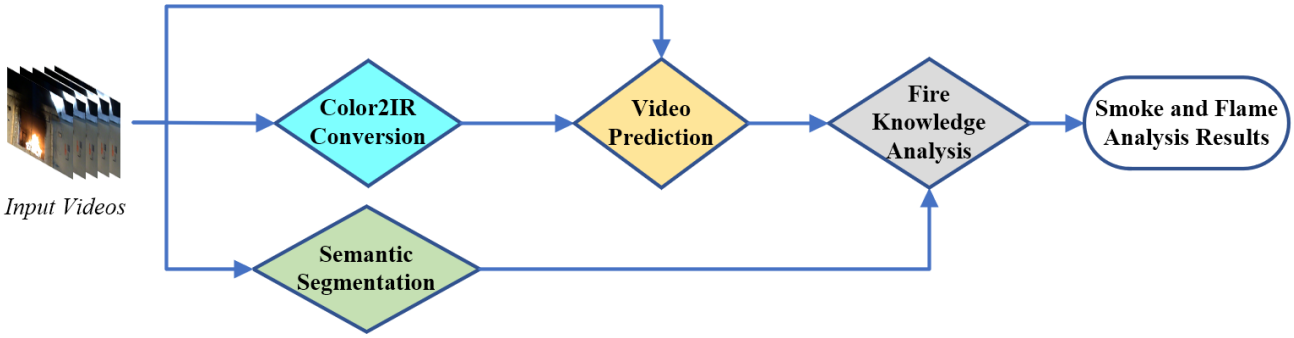


**Figure 1-2:** Overview of our system. The input videos will pass through 4 modules of our system and generate a result of smoke and flame analysis.

The overview structure of our system is shown in **Figure 1-2**. There are four sub-modules in our system. They are the Color2IR Conversion Module, Video Semantic Segmentation Module, Fire Knowledge Analysis Module, and Video Prediction Module. Besides, there is also a fusion part for analysis integration and decision making based on those sub-modules.

The Color2IR Conversion module is a vital part of the process which converts the input vision frames into frames with thermal information (like IR images). The core part of this module is a novel Deep Neural Network that we proposed: DAGAN. It is inspired by cross-domain image conversion methods such as the DiscoGAN [25] and DualGAN [26], Attention mechanism in Computer Vision researches such as, Self-Attention [27], and loop structure that used in CycleGAN [28]. We brought their advantages together and optimized parameters to build a stable and efficient model for Color-to-IR image conversion.

Video Semantic Segmentation Module produces frames with semantic segmentation area information of flame and smoke, which are the most crucial things for fire research analysis. This module is also powered by a Deep Neural Network, TD-Net [29]. We use it as our video semantic segmentation method because it has good accuracy and excellent network efficiency. It takes advantage of Knowledge Distillation [30], which aims to improve an efficient student network trained with a deeper teacher network. It helps a lot in achieving real-time processing while remaining state-of-the-art accuracy.

Video Prediction Module will produce IR or visual images for the future based on RGB and IR frames. This module is built by a Deep Neural Network called Stochastic Adversarial Network (SAN) [31]. We chose this network since it combines the advantages of high-quality output without blurry and diverse predictions. They are both critical as output without blurry could ensure the precision of analysis in the next step of our system, and diverse prediction would give us more space for optimization as the fire phenomenon is complicated for a prediction system without considering

physical constraints.

Then, the Fire Knowledge Analysis Module is built with mathematical models and statistical analysis of flame and smoke areas with temperature information. Although deep learning would help in fire research analysis, experts' fire knowledge and experiences are still important to consider, whose usefulness has been proved [32].

Finally, our system could get the results of flame and smoke analyze from the Fire Knowledge Analysis Module. It could also be further used for several other purposes, such as flame detection, smoke detection, and flashover prediction.

## 1.3 Contributions

For the RGB image-based flame and smoke analysis system proposed by our work, we make contributions as follows:

a) DAGAN for Color2IR Image Conversion
   i. We propose a novel network (DAGAN) that combines two attention modules for background and foreground to generate higher-quality images. And we also design an attention loss in the optimization to solve the saturating problem.

b) Combination of IR and RGB images for image-based flame and smoke analysis
   i. In our system, we combine thermal information from IR images and semantic segmentation information from RGB images in the analysis of flame and smoke.

c) Hybrid design for Image-based flame and smoke analysis
   i. Our system is in hybrid design, combining deep neural networks and fire safety knowledge for real-time flame and smoke analysis with high accuracy.

d) Data augmentation for Video Semantic Segmentation
   i. Data augmentation is applied on the Video Semantic Segmentation Module by introducing synthetic video data for training to achieve high accuracy.

## 1.4 Thesis Organization

In our thesis, Chapter 2 starts with a comprehensive review of Deep Learning, GANs, Autoencoders, and Attention mechanism. Then, we provide a detailed overview of the topics related to sub-modules and our systems, such as Image Conversion, Video Semantic Segmentation, Video Prediction.

Chapter 3 and Chapter 4 are about the work we have done. In Chapter 3, we illustrate the four sub-modules design. We explain the technical details for each sub-module, including network architecture and loss function. We especially emphasized the design of DAGAN for Color-to-IR conversion proposed by ourselves and how we combine the idea of the loop structure, attention mechanism, and GAN to compensate for some specific issues in the Color-to-IR conversion task. Furthermore, we also introduce details on the combination of sub-modules, how we integrate those

four sub-modules together and make them collaborate. Each sub-module performs its duties and acts as a component of the cooperation of all systems. In Chapter 4, we conduct experiments and analyze the results from each sub-module and the whole system for flashover prediction. Then we compare them with other models in performance. We show that our flashover prediction system beat other flashover prediction models both in accuracy and forecast ability.

Chapter 5 summarizes all the works we have done and also concludes our thesis. We also analyze the strengths and weaknesses of our system and then discuss the future work that could improve our system's performance.

# Chapter 2. Literature Review

## 2.1 Deep Learning

Deep Learning is one of the most prevailing research directions of artificial intelligence in recent years. It has been proved to surpass many traditional methodologies that use manual feature extraction in areas around academia and industry. It has boosted the research on a variety of complicated problems.

### 2.1.1 Related knowledge in Deep Learning

Here is some crucial knowledge related to Deep Learning.

#### 2.1.1.1 Convolutional Neural Network

Convolutional Neural Network (CNN) [33] is a typical example of modern Deep Neural Networks. It could extract the features from the local field. This idea comes from the working process of the neurons in the human eyes. Those neurons tend to set up a connection between those pixels of images that is geographically close to each other as they are more likely to share the same features. Another important building block in CNN is kernels (also known as filters). A kernel, or called weights, is just a matrix of values that are trained to detect specific features. The main idea of CNNs is to spatially convolve the kernel matrix on an input image to check if the features are present or not. To provide confidence on how much a specific feature is present, a convolution operation is carried out by computing the kernel's dot product and the input area in overlapped areas. Compared with a fully connected neural network, which was widely used before the 2010s, CNN dramatically reduces the number of parameters and thus reduces the computational cost for both model training and testing.

Nowadays, CNNs are currently widely used in various computer vision tasks, such as classification, detection, and segmentation. The usage of CNNs has significantly progressed with the introduction and development of new and deeper network structures, such as AlexNet [34], ResNet [35, 36], GoogLeNet [37], and VGG-Net [38].

#### 2.1.1.2 Activation function

Activation functions are a type of function that maps the calculated results of the input vector, weights, and bias of neurons to the neuron's output. An essential feature of it is the non-linearity [39, 40], as it aims to map the linear transformation of previous steps in neurons to a high dimensional nonlinear space [41]. Another feature is that they are derivative, which allows the gradients to flow with the networks. The appearance of activation functions is the fundamental reason why Deep Neural Networks could simulate various complex functions in the real world.

Several activation functions are commonly used in research nowadays. The first one is the

Rectified Linear Unit (ReLU). Its definition is shown in equation 2-1.

$$ReLU = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (2-1)$$

ReLU activation function will only keep the positive part of the outputs and generate 0 when the outputs are below 0. Unlike its relatively simple form, it plays a vital role in further promoting the research and development of Deep Neural Networks, especially the deeper ones. The first reason is that it could alleviate the gradient vanishing problem in the calculation process of Deep Neural Networks. Another crucial property of the ReLU activation function is that it makes the neural networks capable of expressing its coefficient.

Sigmoid is another widely used activation function. Its definition is shown in the following equation:

$$Sigmoid = \frac{1}{1 + e^{-x}} \quad (2-2)$$

The sigmoid function is monotonic, and it has a first derivative as bell-shaped. It has two horizontal asymptotes, and it is constrained by them as $x \to \pm\infty$. The sigmoid function is convex when values are less than 0, and it is concave when values are more than 0. As a result, the sigmoid function and its affine compositions can possess multiple optima. In addition, its physical meaning could be translated to the behavior of neuron cell, which means that the output range of (0, 1) could be expressed as the probability that 1 stand for True and 0 stand for False. Besides the advantages listed above, it has its own defect as a gradient vanishing problem. When $x \to \pm\infty$, the derivative of it would also approach 0.

In addition, there are other frequently used activation functions, such as *Tanh*, *LeakyReLU*, and *ELU*. They all play important roles in different scenarios. As a matter of fact, the choice of activation functions has been one of the decisive factors for the performance of Deep Neural Networks as it brings non-linearity to them.

#### 2.1.1.3 Loss function

The loss function acts as the objective of the optimization of neural networks. It maps a network evaluation to a real number, intuitively representing loss value. As the optimization of neural networks would seek to minimize a loss function, the prediction of the model becomes more accurate with the reduction of the loss. The later optimization process will calculate the bias generated from the loss function and feed it into gradients generation for the backpropagation algorithm and update the parameters in the network. The loss function of a neural network can be roughly divided into two categories: Regression loss and Classification loss.

For classification loss, *Cross-entropy Loss (CE)* is a common choice for classification tasks [42]. As shown in equation 2-3, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverge from the actual label.

$$H = -\frac{1}{N}\sum_{i=1}^{N}[y_i log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i)] \quad (2-3)$$

where $N$ is the number of predictions generated from the dataset on all variables, and $y_i$ is the vector of ground truth values of the variable being predicted, while $\hat{y}_i$ is the predicted values.

For regression loss, the two most commonly used loss functions are *Mean Absolute Error (MAE)* and *Mean Square Error (MSE)*. They are also called *L1 loss* and *L2 loss*. The definition of them is shown as the following equation:

$$MAE = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{N} \quad (2-4)$$

$$MSE = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N} \quad (2-5)$$

MAE measures the average of the sum of absolute differences between predictions and ground truth values. It only takes the average magnitude of error irrespective of their direction into account. However, MSE measures the average squared difference between predictions and ground truth values. One thing they have in common is that they both measure the magnitude of error without considering their direction. While, due to squaring factor, predictions that are far away will be more penalized in MSE.

### 2.1.1.4 Backpropagation Algorithm

The Backpropagation Algorithm is one of the decisive factors to the success of Deep Neural Networks. Compared with 'Forward propagate on,' which calculates the results starting from the training data to the final prediction value of the neural network, Backpropagation is used for the optimization of the model. It is based on the 'Chain rule' of derivative, shown as equation 2-6.

$$\frac{\partial F}{\partial x} = \frac{\partial f_2}{\partial f_1} \times \frac{\partial f_1}{\partial x} \quad (2-6)$$

where $F$ is a nested function that $F = f_2(f_1(x))$. In CNN, the derivative of loss of weights is calculated according to the chain rule.

### 2.1.1.5 Optimization

The optimization algorithm is the guidance for updating the weights of the neural network in order to achieve the optimal state defined by the objective function, which contains the set of data batch size, moving average, the learning rate, attenuation, and other strategies to speed up the optimization and iteration of the neural network [43]. During this process, the loss function and gradient guide the right direction for the optimizer to move forward [44].

Optimizers like SGD [45], RMSProp [46], SGD with Momentum [47], AdaGrad [48], and Adam [49] have shown significant ability to effectively decreasing the gradient in recent studies. Adam optimizer is the most popular one among them for its stability and adaptive feature.

Adam could be somehow regarded as a combination of RMSprop and SGD with momentum methods. It uses the squared gradients in the learning rate scaling just like RMSprop, while simultaneously taking advantage of the momentum with the moving average of the gradient other than the gradient itself [43, 49]. Furthermore, Adam is an adaptive learning rate method, and it computes learning rates for different parameters individually. It uses estimations of the first and second moments of the gradient to adapt the learning rate for each weight of the neural network [44].

### 2.1.2 Application in Fire Research Field

As a type of novel technology in the Computer Vision field, deep learning also arouses the interest of researchers in other fields at the same time. Deep Learning was brought in this area with curiosity and excitement as a field where traditional manual feature extraction approaches are widely used. Many studies have achieved good results in some general tasks for fire research, such as fire detection, smoke detection, and fire prediction [6]. Some of them even greatly surpass the performance of traditional methods.

#### 2.1.2.1 Fire Detection

As one of the inevitable themes of fire research, fire detection (sometimes referred to as flame detection or fire flame detection) is one of the first tasks that has already introduced Deep Learning. In [50], Xu et al. propose a fire detection method based on color features, wavelet analysis, and CNN. It is the first appearance of CNN in fire flame detection, and it was only used as a complementary method for decision making. After that, Zhong et al. in [51] propose a novel flame detection algorithm based on a Deep CNN. It achieves outstanding performance with an accuracy rate in experiments reaching 97.64% and far surpasses other methods. Then, Kim et al. [52] brought in Faster R-CNN for the same task and achieved a comparable accuracy with a much higher detection speed. Besides, Filonenko et al. propose a combination of a CNN and recurrent neural network (RNN) to detect the fire flame in space and time domains [24]. The CNN automatically builds the low-level features, whereas the RNN finds the relation between the features in different frames of the same event.

#### 2.1.2.2 Smoke Detection

Smoke detection is another important task in the field of fire research. As smoke could sometimes not be visible to human eyes and cameras, and it would also appear earlier than fire flame, smoke detection has attracted many researchers' attention. It is also an area in Deep Learning that has been widely used. In [19], Zhang et al. propose a Dual-Channel Convolutional Neural Network (DC-CNN) using transfer learning for detecting smoke images. Their network achieved detection of over 99.33% on a publicly available dataset. Hu et al. brought in Spatio-Temporal CNN for a video smoke detection in [53]. Their model achieves a 97% detection rate with a 3.5% false alarm rate in a customed dataset. In addition, Yin et al. proposed a method based on RNN in [23]. They train the model to capture the space and motion context information in smoke videos. The network model has a high performance achieving true positive rates of over 95% and true negative rates of over 97%.

#### 2.1.2.3 Fire Prediction

The prediction of the fire phenomenon is a crucial part of fire analysis, evacuation plan, and fire service decision-making. A variety of models have been built to monitor a sudden fire propagation called flashover. In [54], Fu et al. built a flashover prediction model, which can be used to warn firefighters before flashover occurs. Overall validation shows that their model's prediction accuracy is around 75%. Some researchers also tried to combine multiple factors with improving the performance of their model. In [55], Yap et al. introduce a model based on the Generalized Adaptive Resonance Theory (GART) neural network developed based on integrating Gaussian ARTMAP and the Generalized Regression Neural Network. Their model takes both temperature and Heat Release Rate (HRR) as input and gives binary prediction for flashover. Their model outperforms other networks and produces meaningful rules from data samples.

The most related paper to our work is the one proposed by Yun et al. [56]. It is also an end-to-end system for flashover prediction with image conversion. They focused on temperature analysis and applied Generative Adversarial Neural Networks for image conversion and enhancement, from RGB images to IR images. Then, they analyze the temperature variation only based on the information from converted IR images in several ranges and predict the flashover, which could be as early as 55 seconds before flashover occurs.

## 2.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs) is a group of Deep Learning frameworks that were first introduced by Goodfellow et al. in 2014 [57]. It is applied as an unsupervised model, which opened up a new idea in the field of data generation. It could easily approach a specific distribution of the dataset given in the training process. In addition to the areas mentioned above, GANs have also been proved effective in supervised learning and semi-supervised learning, such as representation learning [58], image inpainting [59], neural style transfer [28, 60], and image super-resolution [61, 62], etc.

### 2.2.1 General Principle

The GANs have two individual parts of neural networks, which are in an adversarial relationship. It is inspired by the two-person zero-sum game, and it can achieve the best generation effect by two networks confronting each other and achieving the Nash equilibrium in the training process. The two parts are called generator ($G$) and discriminator ($D$). Unlike other Neural Networks with a strict definition for network structure, GANs have no restriction in their early proposed time. Researchers are also diverging their thinking and have established various types of GANs for different scenarios. An example of the general architecture of GANs is shown in **Figure 2-1**.

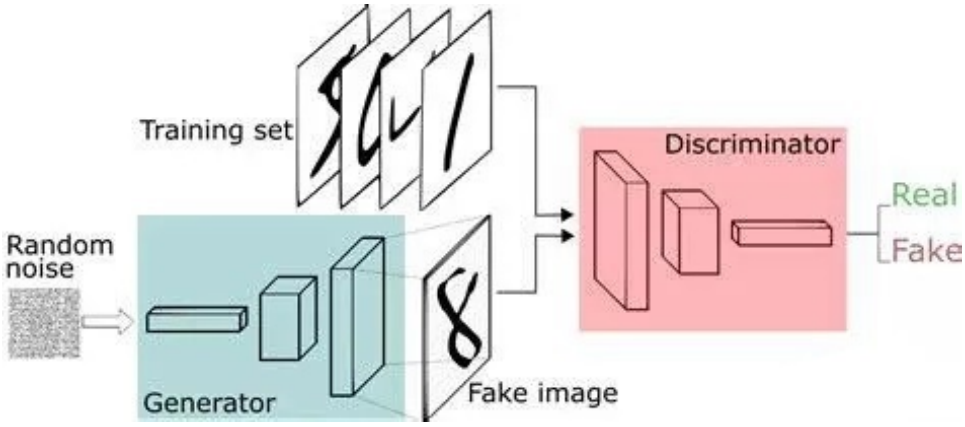


**Figure 2-1:** The general architecture of GANs, and the principle of data process in it.

In general, the generator $G$ aims to generate 'fake' data that fir the distribution of real data in the training set as much as possible, while for the discriminator $D$, it needs to distinguish the real data in training set with the 'fake' data generated by generator $G$.

For the generator $G$, it starts with the random noise vector $z$. Those noises are usually set to follow the Gaussian or uniform distribution. Then the noise vector will be fed into generator $G$, and it will be mapped to a fake data sample $G(z)$. $G(z)$ is in data space, so generator $G$ is doing mapping work from noise sample space $Z$ to data sample space $X$.

While, for the discriminator $D$, it takes a pair of generated fake samples and real samples from the training set as input. The output of it will be the probability of the classification result of a generated fake sample that whether it is real or not. As a matter of fact, discriminator $D$ works as a binary classifier. It is also the state indicator for the optimization process of GANs. If the discriminator $D$ could exactly tell whether the data is a generated one or a real one, we would consider that the optimal state of GANs training is reached. Generator $G$ would also learn the distribution of real data at that time.

As an unsupervised training process, the loss function of GANs contains Jensen-Shannon divergence between the real data distribution $\mathbb{P}_r$ and prior distribution $\mathbb{P}_g$. The overall loss function is shown as follow:

$$\min_G \max_D \mathbb{E}_{x\sim\mathbb{P}_r(x)}\left[\log\left(D(x)\right)\right] + \mathbb{E}_{x\sim\mathbb{P}_g(x)}\left[\log\left(D(\tilde{x})\right)\right] \qquad (2-7)$$

where $x$ is sampled from the real data distribution $\mathbb{P}_r(x)$, and $\tilde{x}$ is sampled from the prior distribution $\mathbb{P}_g(x)$ where Gaussian or uniform distribution is usually used. $\mathbb{E}(\cdot)$ is the mathematical expectation.

Equation 2-7 is the objective for minimizing. It could be a deduction from mathematical principles that it will achieve its minimum value when:

$$D_g^*(x) = \frac{\mathbb{P}_r(x)}{\mathbb{P}_r(x) + \mathbb{P}_g(x)} \qquad (2-8)$$

It is the optimal solution of discriminator $D$.

Though it might seem to be easy and comfortable to optimize a GAN theoretically, the actual

operation process is often more complicated as it might quickly fail to be optimal when one part of the GAN is too strong and the other one is weak. As a matter of fact, we usually use an alternative training method. First, we fix $G$ and optimize $D$ for maximum discriminator accuracy. That is also called 'buffer' for $D$. Then, we fix $D$ and optimize $G$ to minimize discriminator accuracy. These two steps will alternate and only stops when $\mathbb{P}_r = \mathbb{P}_g$. They form a close loop, and we will run this one until optimal.

Since 2014 when GANs were introduced, researchers have highlighted improved ideas for all aspects of GAN, such as training stability and scope of application. Here we introduce one of the most used ideas in GAN improvement, and it is also related to our studies.

To enhance the stability of GANs, Wasserstein GAN (WGAN) [63] is proposed, and it solves the problem of training optimization easily with the modification of divergence. The Jensen-Shannon (JS) divergence of the objective function is a great choice to measure the divergence between distributions for most of the time. While, in some cases, the distribution of real data and prior (fake data) could have very little overlap. Furthermore, it could even be a constant in extreme cases and thus cause the gradient vanishing problem in the optimization process. The author of WGAN uses Earth-Mover (SM) distance instead of the JS divergence to evaluate the distance of distribution between real and the generated data, which could stabilize the optimization process. Compared with JS divergence, Wasserstein distance comes from the idea of measuring the distance on earth, which is more stable and avoids the gradient vanishing problem by still providing a meaningful gradient in the calculation. The equation for calculating Wasserstein distance is shown as follow:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \prod(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|] \qquad (2-9)$$

where $\mathbb{P}_r$ is the data distribution, $\mathbb{P}_g$ is the greatest lower bound for any transport plan in mathematic. $\prod(\mathbb{P}_r, \mathbb{P}_g)$ is the set of all joint distributions $\gamma(x, y)$, whose marginals are $\mathbb{P}_r$ and $\mathbb{P}_g$.

While the equation 2-9 might be perfect in mathematics. However, it is intractable for optimization usage. We can simplify it by using the Kantorovich-Rubinstein duality, and it will become to equation 2-10 as follow:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_g}[f(x)] \qquad (2-10)$$

where $sup$ is the least upper bound and $f$ is a 1-Lipschitz function following the constraint: $|f(x_1) - f(x_2)| \leq |x_1 - x_2|$.

Conditional GAN (cGAN) [64] is another vital derivative of GANs as it provides an additional control factor for GANs. An illustration of the cGAN structure is shown in **Figure 2-2**.

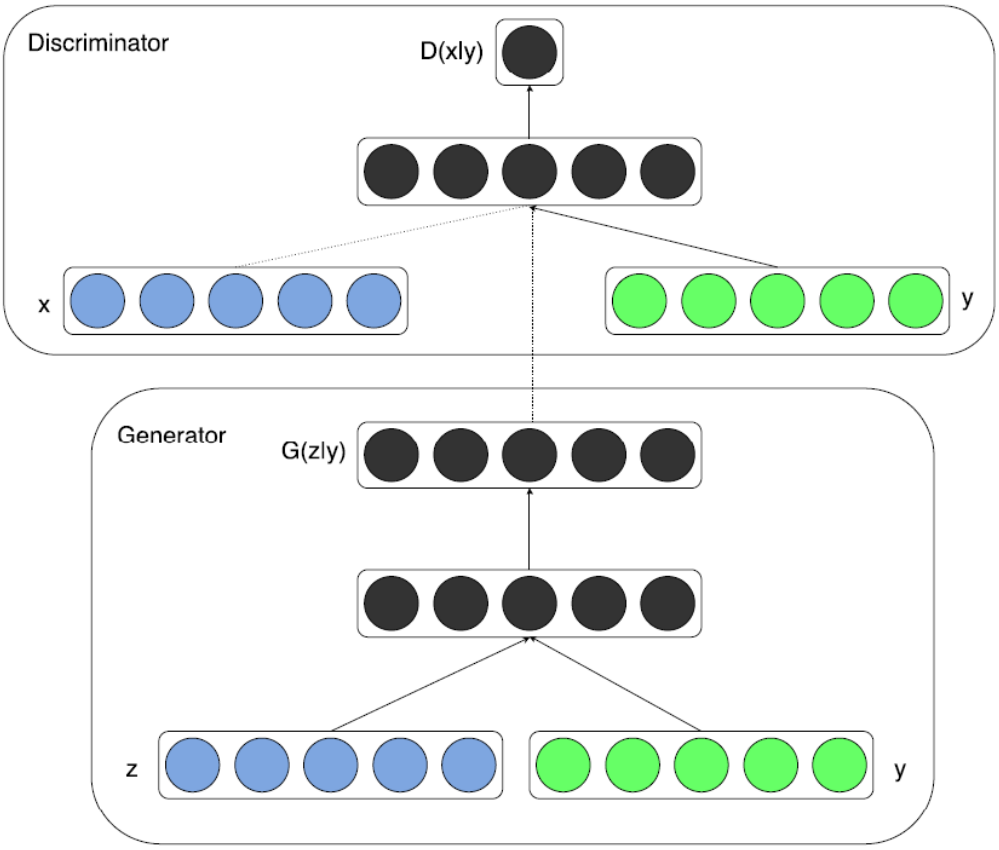


**Figure 2-2:** An illustration of the cGAN structure, from [64].

The optimization loss function of cGAN is shown in the equation below.

$$\min_{G}\max_{D}\mathbb{E}_{x\sim\mathbb{P}_r(x)}\left[\log\left(D(x|y)\right)\right]+\mathbb{E}_{z\sim\mathbb{P}_z(z)}\left[\log\left(1-D\left(G(z|y)\right)\right)\right] \qquad (2-11)$$

where $x$ is sampled from the real data distribution $\mathbb{P}_r(x)$, $y$ is the conditional variable and $z$ is sampled from the prior distribution $\mathbb{P}_z(z)$.

The general structure is almost the same as the original GAN. At the same time, the introduction of conditional variable $y$ marks the start of modifying GANs for unsupervised learning to supervised and semi-supervised learning. It has inspired a lot of future works since then.

### 2.2.2 GANs for Image Conversion

Image conversion, sometimes also referred to as image transfer, is a type of task that deals with the conversion of image pairs from one to the other. An example of image conversion tasks is shown in **Figure 2-3** below.


**Figure 2-3:** An example of image conversion tasks. (the first row)

Researchers have made significant achievements in this field. This task can be roughly divided into two categories, depending on the conditions of input image pairs. If they are completely aligned, which means that the semantic of one pixel for one image is related to the corresponding pixels in the other one at the same location, it would be called paired or pixel-to-pixel (pix2pix) task. The image below shows a comparison of paired image conversion tasks and un-paired image conversion tasks.

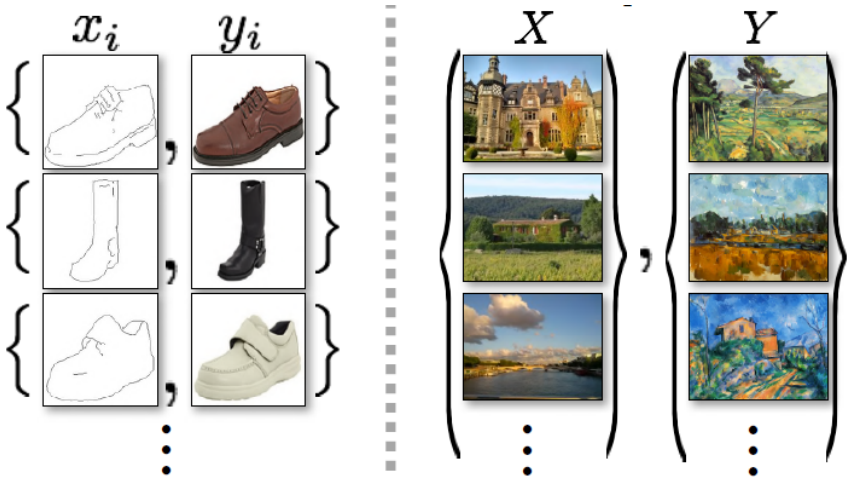


**Figure 2-4:** A comparison of paired and unpaired image conversion tasks, (left: paired conversion task; right: un-paired conversion task).

For pix2pix tasks, cGAN is a typical choice for the GANs selection. An illustration of using cGAN in pix2pix tasks is shown in Figure 2-4 below.

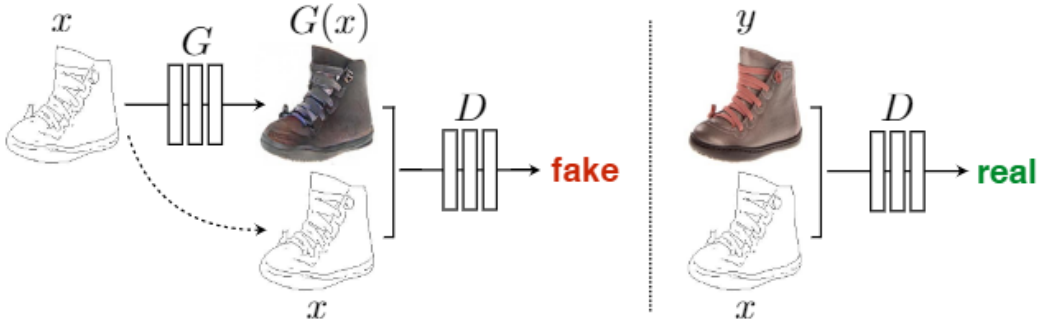


**Figure 2-5:** An illustration of using cGAN in pix2pix tasks, from [60].

The conditional variable $y$ is a vital part of this system. As the path to introduce conditional information in the learning process, if we put suitable supervised information in this path, it should make the generator $G$ produce results similar to the conditional input.

For the example in **Figure 2-5**, The discriminator, $D$, learns to classify between fake and real $\{\text{edge}, \text{photo}\}$ tuples. The generator, $G$, learns to fool the discriminator. Unlike unconditional GANs, both the generator and discriminator could observe the input edge map.

Although cGAN has an excellent performance in the pix2pix task, when the input image pair is not strictly aligned, the image it generates will have serious problems, such as blurry and color shift. As a matter of fact, most of the data collected in the real world is not perfectly aligned, just like no one could find two same leaves on the tree. Though it might be alleviated by data preprocessing to some extent, researchers have proposed a new solution for these kinds of tasks, known as un-paired image conversion.

DiscoGAN is one of the first new GAN structures proposed aiming to solve these tasks. It could successfully transfer style from one domain to another and, meanwhile, preserve other attributes in the background.

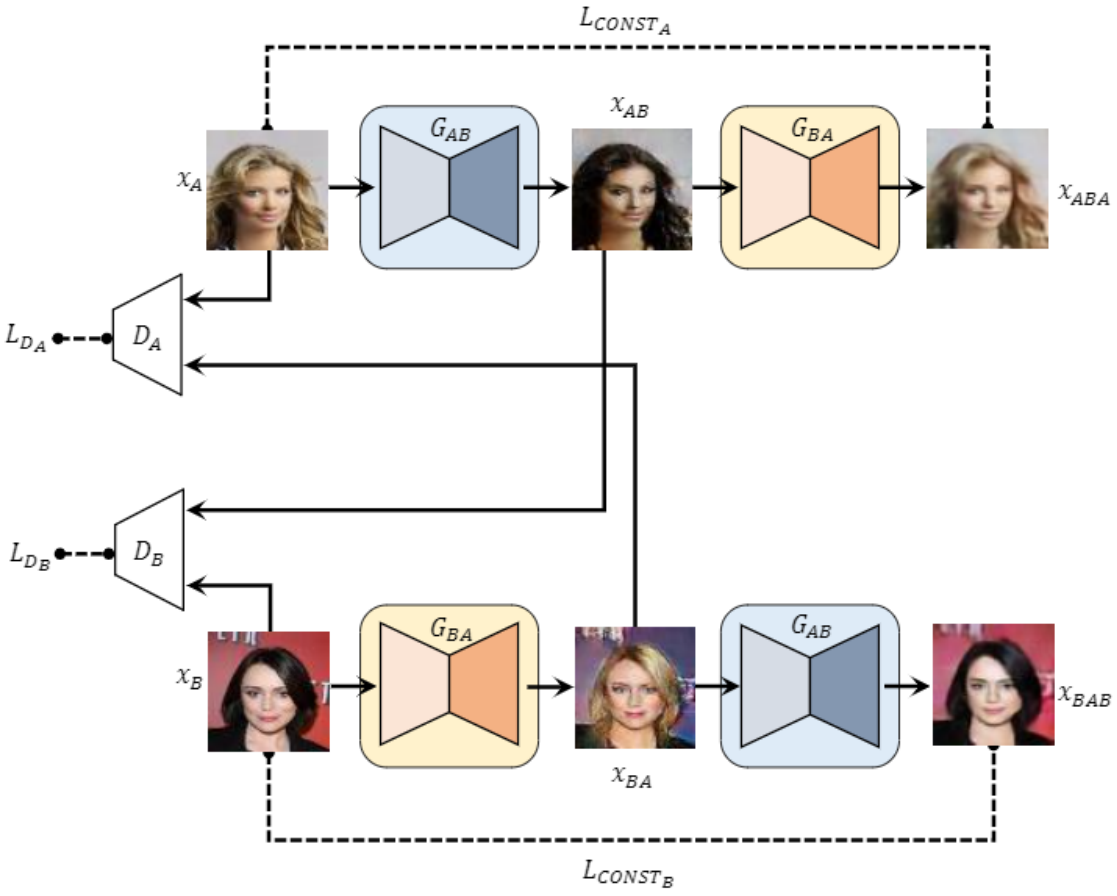


**Figure 2-6:** Architecture of DiscoGAN, from [25].

Shown as the **Figure 2-6** above, DiscoGAN has two generators $G_{AB}$, $G_{BA}$ and two

discriminators $D_A$, $D_B$. Generators $G_{AB}$ will translate the input $X_A$ to $X_{AB}$, while Generators $G_{BA}$ will have an inverse process, that is, translate the input $X_{AB}$ to $X_{ABA}$. Then, the $X_{ABA}$ will be used to match the input $X_A$ since they are in the same domain. The mathematical process of it is shown as follow:

$$X_{AB} = G_{AB}(X_A) \tag{2-12}$$

$$X_{ABA} = G_{BA}(X_{AB}) = G_{BA}\big(G_{AB}(X_A)\big) \tag{2-13}$$

The mechanism that helps them succeed in those un-paired image conversion tasks is the loss function design. They introduce $L_{const_A}$, which is defined by the difference of reconstruction image in original domain and the original input image. The mathematical definition is shown as follows.

$$L_{const_A} = d\big(G_{BA}\big(G_{AB}(X_A)\big), X_A\big) \tag{2-14}$$

where $d(\cdot)$ is a function that measures the distance between the reconstructed image and the original one. It could be a pixel difference.

They add this $L_{const_A}$ to the optimization function of generator training. The loss of total generator and discriminator is formulated as follows.

$$L_G = L_{G_{AB}} + L_{G_{BA}} = L_{G_{ANB}} + L_{const_A} + L_{G_{BNA}} + L_{const_B} \tag{2-15}$$

$$L_D = L_{D_{AB}} + L_{D_{BA}} \tag{2-16}$$

CycleGAN [28] is a further improved model for un-paired image conversion tasks based on DiscoGAN. It adopts GAN with a new type of loss called: cycle consistency loss. An illustration of CycleGAN architecture is shown in **Figure 2-7**.

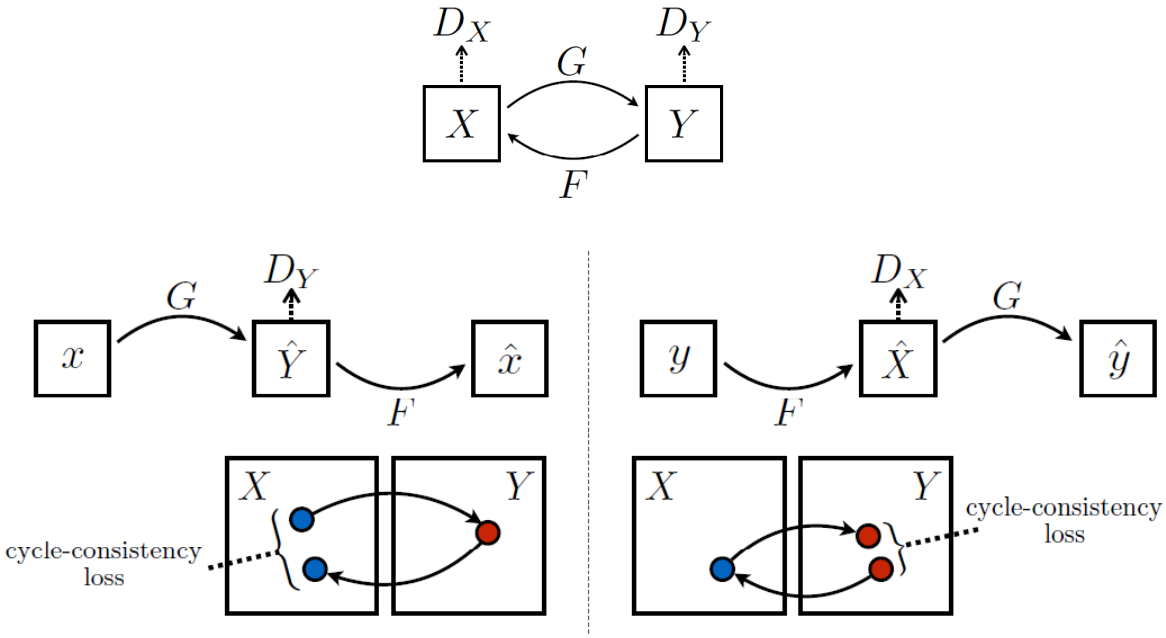


**Figure 2-7:** Architecture of CycleGAN, and a detailed version of forward and backward process of conversion between 2 domains, from [28].

The cycle consistency loss of CycleGAN is similar to the reconstruction loss of DiscoGAN. For the mapping function $G$, the objective is expressed as follows.

$$L_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim P_{data}(y)}[\log(D_Y(y))] + \mathbb{E}_{x \sim P_{data}(x)}[\log(1 - D_Y(G(x)))] \quad (2-17)$$

Thus, cycle consistency loss can be formulated as the following equation.

$$L_{cyc}(G, F) = \mathbb{E}_{x \sim P_{data}(x)}\left[\left\|F(G(x)) - x\right\|_1\right] + \mathbb{E}_{y \sim P_{data}(y)}\left[\left\|G(F(y)) - y\right\|_1\right] \quad (2-18)$$

Then, the overall loss function for generator and discriminator is:

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F) \quad (2-19)$$

where $\lambda$ is a parameter set for optimization.

In order to further improve the image conversion quality for un-paired image conversion tasks, recent research introduces a method commonly used in Neural Language Processing (NLP) tasks. It is the Attention mechanism. The principle of the attention mechanism comes from human visual neurons. When the human eye views an object, although a large amount of object information is taken in, the human brain can freely choose the attention area. Only the visual information in the attendance area will be sent to the optic nerve cells to activate the neurons. This is also the reason why people are distracted. Inspired by this idea, researchers hope to use the attention mechanism to distinguish the transition areas that need to be focused, and at the same time, try to keep the non-key areas unchanged during the transition. Figure 2-8 shows the general architecture of their idea: Attention-guided GAN (AGGAN).

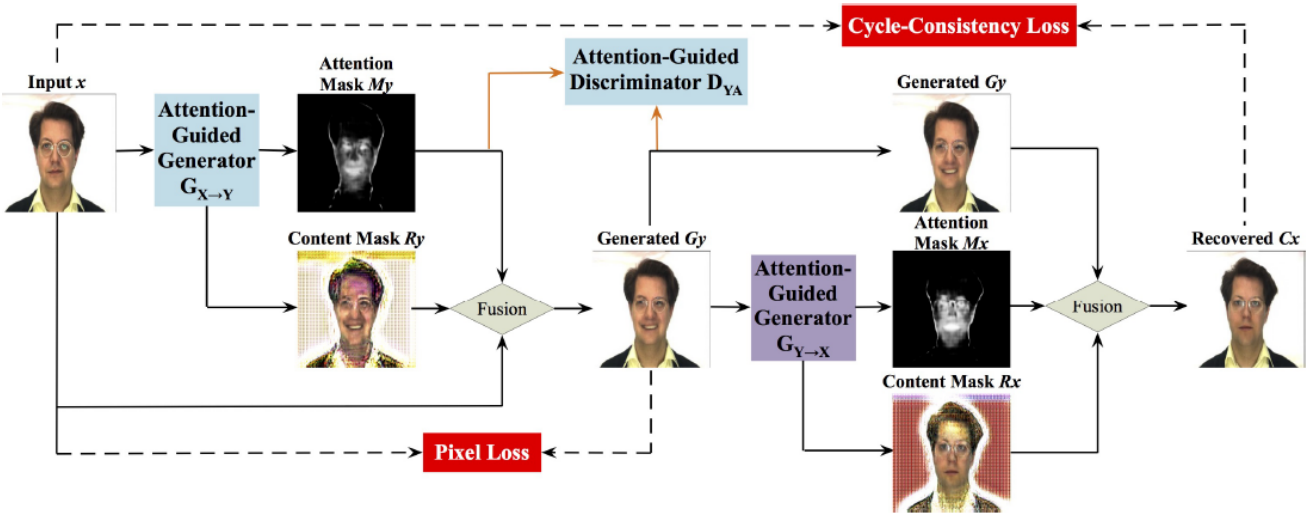


**Figure 2-8:** Architecture of AGGAN, from [65]. It uses foreground attention in the generation and reconstruction process. It uses pixel loss and cycle-consistency loss for loss function.

The structure of the generator of AGGAN, $G_{X \to Y}$ is similar to that of CycleGAN, while the difference is the output layer. Three original channels of output would remain the same as the content mask $R_y$. And, there is another output channel that would go in *Sigmoid* activation function and form a vector called attention mask $M_y$. Then, images $G_y$ will be generated following the equation below.

$$G_y = R_y \times M_y + x \times (1 - M_y) \quad (2-20)$$

where $x$ is the input image from one domain.

For the discriminator, they have two individual ones in this process. $D_Y$ is similar to a normal discriminator as it would distinguish fake generated examples and real examples. $D_{YA}$ is a brand new discriminator that takes the attention mask as conditional input. So, the input vector of $D_{YA}$ should be $[M_y, y]$, which is a 4-dimension vector. Same as the CycleGAN, $G_{Y\to X}$ , $D_X$ and $D_{XA}$ have a similar structure as $G_{X\to Y}$, $D_Y$ and $D_{YA}$, while the input and output of them are swapped.

Besides, they also introduce Attention-Guided Adversarial Loss $L_{AGAN}(G_{X\to Y}, D_{YA})$ in the loss function. It is an adversarial loss that takes the attention mask $M_y$ as conditional input for better optimization. It could be expressed as the following equation.

$$\begin{aligned}\mathcal{L}_{AGAN}(G_{X\to Y}, D_{YA}) = {} & \mathbb{E}_{y\sim P_{data}(y)}\big[logD_Y\big([M_y, y]\big)\big] \\ & + \mathbb{E}_{1-y\sim P_{data}(x)}\big[log(1 - D_Y\big([M_y, G_{X\to Y}(x)]\big))\big] \end{aligned} \quad (2-21)$$

Then, they add it to the final loss function, and other parts are similar to CycleGAN. It could be formulated as equation 2-22.

$$\begin{aligned}\mathcal{L}(G_{X\to Y}, G_{Y\to X}, D_X, D_Y, D_{XA}, D_{YA}) = {} & \lambda_{GAN}[\mathcal{L}_{GAN}(G_{X\to Y}, D_Y) + \mathcal{L}_{GAN}(G_{Y\to X}, D_X) \\ & + \mathcal{L}_{AGAN}(G_{X\to Y}, D_{YA}) + \mathcal{L}_{AGAN}(G_{Y\to X}, D_{XA})] \\ & + \lambda_{cycle} \times \mathcal{L}_{cycle}(G_{X\to Y}, G_{Y\to X}) \\ & + \lambda_{pixel} \times \mathcal{L}_{pixel}(G_{X\to Y}, G_{Y\to X}) \end{aligned} \quad (2-22)$$

where $\lambda_{GAN}$, $\lambda_{cycle}$, $\lambda_{pixel}$ are parameters that set for optimization.

# 2.3 Autoencoders

Autoencoders (AE) [66] is a structure that contains encoder and decoder parts. Modern autoencoders are made of neural networks. Autoencoders are one of the earliest structures that are made to learn data coding in an unsupervised manner with efficiency. It could be briefly divided into three parts: input layers, hidden layers, and output layers. Autoencoders have been proved useful for various usage, including encoding and decoding, dimensionality reduction, and feature extraction. Researchers have also made significant progress and produced various variants such as Variational Autoencoder (VAE) [67], Adversarial Autoencoder (AAE) [68], Denoising Autoencoder (DAE) [69], etc.

## 2.3.1 Autoencoders

**Figure 2-9** shows the structure of the most basic autoencoders. It starts with an input layer, then to a latent layer with a relatively low dimension, and then to the decoder structure to reconstruct the results. These kinds of hidden with low dimensions are also called ‘bottleneck.’ Autoencoder aims to minimize the difference or error between the input and reconstructed output.

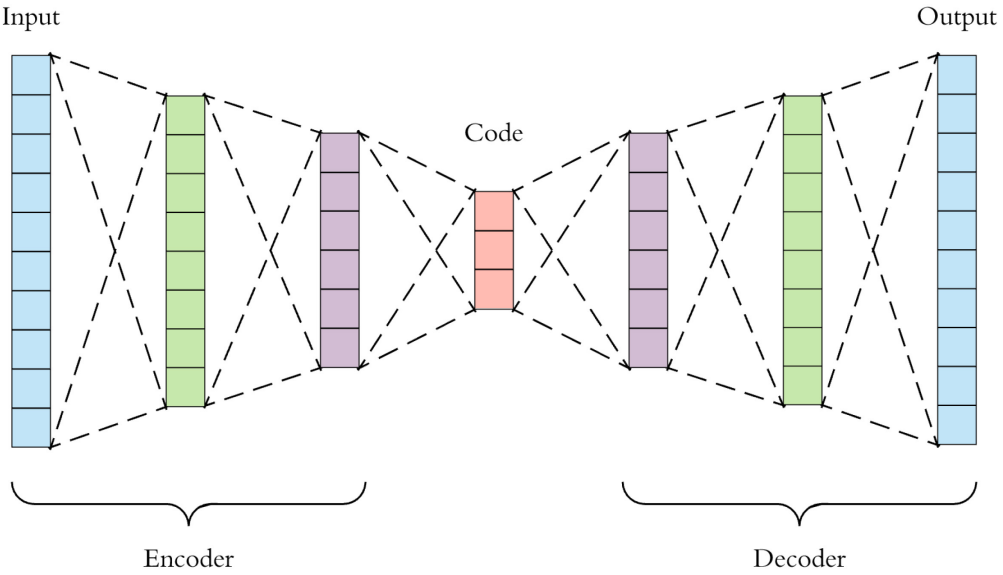


**Figure 2-9:** The general architecture of autoencoders.

Besides, there is another encoding method in the traditional mathematical area: Principal Component Analysis (PCA) [70]. Autoencoders are closely related because they both can reduce dimensionality in an unsupervised manner. PCA could be regarded as an autoencoder that uses linear activation functions. Compared with PCA, autoencoders usually have better performance in feature extraction resulting from the attribute of deep neural networks.

### 2.3.2 Variational Autoencoders

Unlike AE for encoding and compression purposes, VAE [67] is made for a generative purpose. In order to be able to use the decoder of our autoencoder for generative purposes, the latent space needs to be regular enough. As a result, regularization is an essential part of the VAE. It needs to avoid overfitting problems and ensure that the latent space has good properties to enable the generative process.

Since VAE is for generative purposes, the target of it should be changed to a fixed vector to a distribution $p_\theta$. It could learn the prior $p_\theta(z)$, posterior $p_\theta(z|x)$ and likelihood $p_\theta(x|z)$ from the mapping of input to the latent space.

For regularization term, it is expressed as the Kullback-Leibler (KL) divergence between posterior $q_\phi(z|x)$ and the real distribution $p_\theta(z|x)$, which is:

$$D_{KL}\left(q_\phi(z|x)\middle|p_\theta(z|x)\right) \qquad (2-23)$$

## 2.4 Attention Mechanism

### 2.4.1 Origins of Attention Mechanism

Attention mechanism comes from the tasks in the NLP area. Its core logic is to change from focusing on everything to focusing only on the vital part, that is, picking the critical part. It comes from some state-of-the-art models in the NLP area. Bidirectional Encoder Representations from Transformers (BERT) [71] and Generative Pre-trained Transformer (GPT) [72] are famous pre-trained models proposed by Google. The same thing between those two models is that they all make use of the transformer part. Inside the transformer architecture, the attention mechanism is the core part that makes the transformer accurate and efficient for NLP tasks.

Attention mechanisms have several significant advantages. The first one is fewer parameters. Compared with CNN and RNN, the model complexity would be minor, and the parameters are also fewer. Therefore, the computing costs would also be smaller. The second reason is that it is fast. Attention solves the problem that RNN cannot be calculated in parallel. Moreover, each step of the Attention mechanism does not depend on the calculation results of the previous step, so it can be processed in parallel like CNN, which could dramatically reduce the time cost. The third one is excellent performance in NLP tasks. Before the introduction of the Attention mechanism, there was a problem that everyone had been distressed: long-distance information will be weakened, just like a person with weak memory ability cannot remember the past. The emergence of the Attention mechanism perfectly solves this problem.

There are three main components in the attention mechanism: Query ($Q$), Key ($K$), and Value ($V$). Key and Values come from the source. Query is the area that we are interested in, while Key and Values are pairs that are stored in Source where we are looking for additional information. This process could be regarded as looking for a book when we only know a topic that we are interested in. Values represent the books, and Keys are the ID numbers for them.

After a brief idea of how the attention mechanism works, let us define it mathematically. **Figure 2-10** shows an example of Attention Value calculation with 4 Keys and Values.

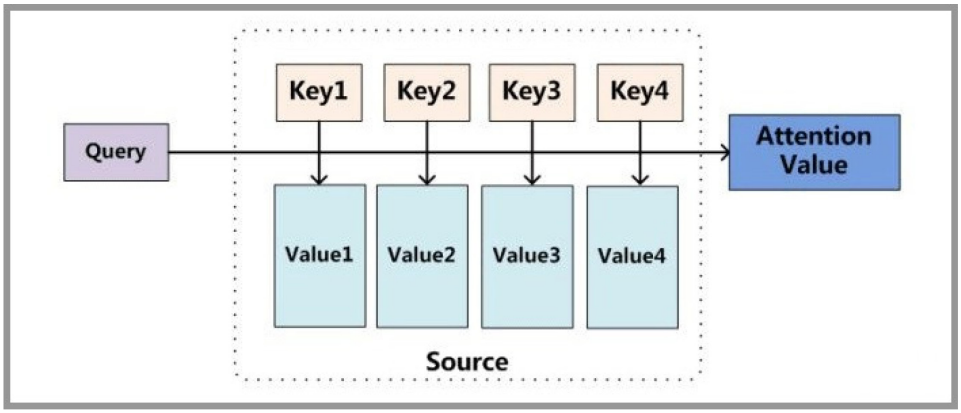


**Figure 2-10:** An example of Attention Value calculation. The Query calculates attention value with 4 Keys and Values.

It includes three steps. The first one is to calculate the similarity between Q and K to get the weights. There are several similarity calculation methods, including dot product (or matrix multiplication), cosine similarity, and simply splice them together, shown as the equation 2-24.

$$s(q,k) = q^T k$$

$$s(q,k) = \frac{q^T k}{\|q\| \times \|k\|}$$

$$s(q,k) = W[q;k] \qquad (2-24)$$

While the weights calculated directly from the first step could not be put into further calculation because it has not yet been normalized, the goal of the second step is to do normalization for those results and get a usable weight for further process. A typical choice for it is the SoftMax function.

After that, we could calculate the weighted sum of the V and corresponding weights, as in the equation below.

$$A_N = \sum_{i=1}^{N} W_i \times V_i \qquad (2-25)$$

Those are the three steps for attention value calculation.

### 2.4.2 Attention in Computer Vision

The success of the Transformer module in the NLP area has drawn the attention of researchers in the Computer Vision area. As attention mechanism also happens in visual aspects of human neurons, researchers are confident that the transformer module, especially the attention mechanism, would work in Computer Vision tasks, such as classification, detection, and segmentation.

Attention module was first proposed in Computer Vision tasks in 2017 [27]. They took the idea of soft attention and built two types of self-attention modules in their works. Their structure is shown in **Figure 2-11**.

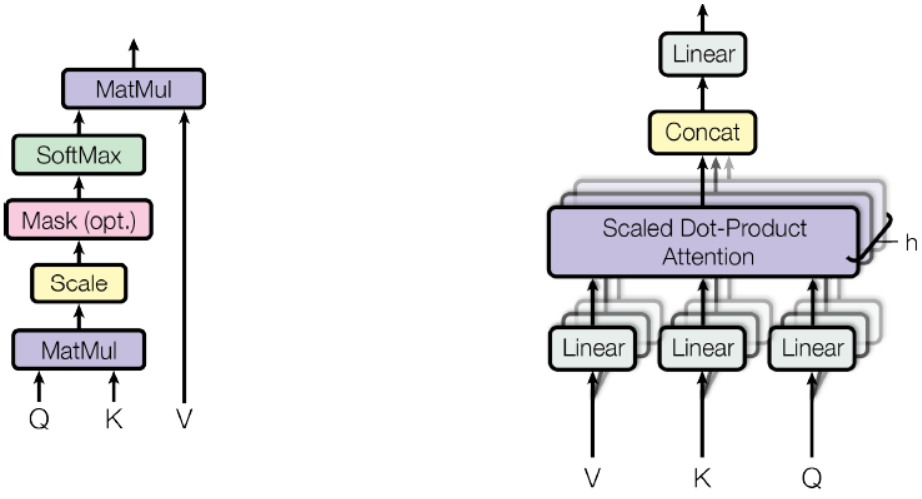


**Figure 2-11:** Structure of 2 self-attention modules (left: Scaled Dot-Product Attention, Right: Multi-Head Attention), from [27]

Scaled Dot-Product Attention is a typical example of a single-attention module. It performs attention with keys, values, and queries with the same and specific dimensions. The calculation could be formulated as below.

$$\text{Attention}(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (2-26)$$

where $Q, K, V$ are the matrices that queries, keys and values are packed up together. $d_k$ is the dimension of $K$.

This attention module is formulated based on Dot-product attention. While they also added a scaling factor $\frac{1}{\sqrt{d_k}}$, which could dramatically alleviate the magnitude growth of $QK^T$, that could push $softmax$ function to regions with extremely small gradients [73].

The second type of attention module is Multi-Head Attention. It is generally similar to Scaled Dot-Product Attention, while it performs the calculation in parallel. It first projects keys, values, and queries linearly. Then they perform the attention function in parallel. Finally, they concatenated and once again projected, resulting in final values, as depicted in **Figure 2-12**. The calculation process is shown in the equation below.

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(head_1, \dots, head_h)W^O \\ \text{where } head_i &= \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \end{aligned} \qquad (2-27)$$

After the publication of this article, the application of the attention mechanism in the Computer Vision field has sprung up in recent years. Among these applications, the combination of attention mechanism and GANs is particularly outstanding. There are excellent works, such as Self-Attention GAN (SAGAN) [74], Attention-Guided GAN (AGGAN) [65]. Some of them even surpassed the state-of-the-art approaches in basic tasks in Computer Vision, such as classification, detection, and segmentation.

# 2.5 Video Semantic Segmentation

## 2.5.1 Image Semantic Segmentation

As one of the fundamental tasks in the Computer Vision field, segmentation, especially semantic segmentation, plays a broad role in various applications, including scene understanding, medical image analysis, robotic perception, video surveillance, augmented reality, and image compression, among many others [75]. Numerous approaches have been developed, from the earliest methods, such as thresholding [76], histogram-based bundling, region-growing [77], and k-means clustering. Over the past few years, Deep Learning has yielded a new generation of segmentation with remarkable performance improvement.

The image segmentation task could be formulated as a classification problem for pixels of semantic labels. It could perform pixel-level labeling with several categories. An example of semantic segmentation is shown in **Figure 2-13**.

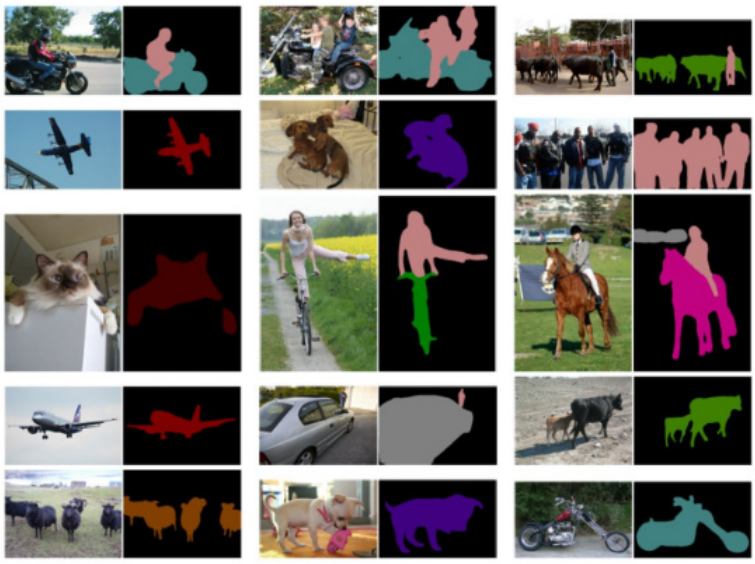

**Figure 2-12:** Semantic segmentation results. For each image pair, the left one is the original image and the right one is the segmentation results.

There are several categories of Deep Learning models proposed for semantic segmentation in recent years. Here we introduce some of them.

The first approaches with Deep Learning for semantic segmentation are proposed by Long et al. in [78]. They called it a Fully Convolutional Network (FCN). **Figure 2-13** shows that it only uses the convolutional layers to produce a segmentation map. They also modified the existing CNN models, such as VGG16 and GoogLeNet, to build a non-fixed input and output with fully convolutional layers.

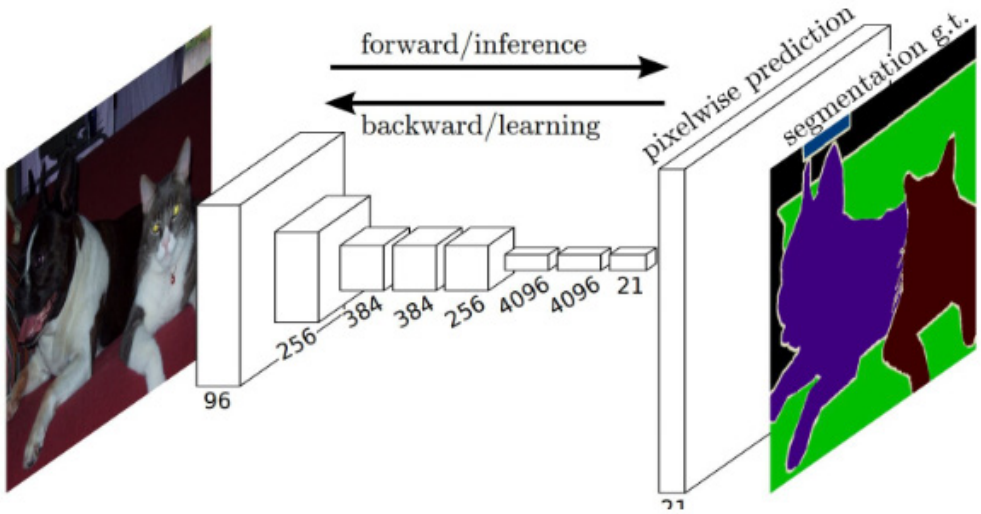


**Figure 2-13:** The structure of Fully Convolutional Network (FCN), from [78]. It only uses the convolutional layers to produce a segmentation map. They also modified the existing CNN models, such as VGG16 and GoogLeNet, to build a non-fixed input and output with fully convolutional layers

FCN is always considered a significant milestone for image segmentation, proving that Deep Neural Networks could be trained for end-to-end segmentation. While, as an old approach, it also has its shortcomings, such as it could be used for real-time inference, and it is also not efficient.

Another popular family for image segmentation is encoder-decoder-based models. They are efficient for pixel-wise semantic segmentation.

The deconvolution function was proposed in solutions for semantic segmentation tasks by Noh et al. in [79]. Their model consists of 2 parts. The first one is an encoder with convolutional layers adopted from VGG16 models. The second part is a deconvolutional network that generates a map prediction for pixel semantic. The deconvolution network is built of deconvolution and Unpooling layers, Shown in **Figure 2-14**.

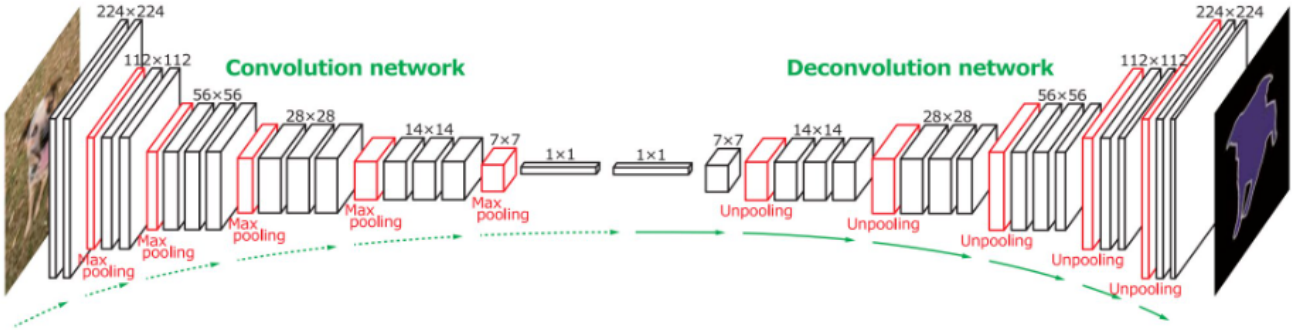


**Figure 2-14:** The structure of Deconvolutional semantic segmentation, from [79]. It consists of 2 parts. The first one is an encoder with convolutional layers adopted from VGG16 models. The second part is a deconvolutional network that generates a map prediction for pixel semantic.

Pyramid network is also a typical structure in semantic segmentation models. Pyramid Scene Parsing Network (PSPNet) was developed by Zhao et al. in [80]. It uses a residual network (ResNet) as the backbone for feature extraction. These feature maps are then fed into a PSP module for pattern distinction in various scales, shown in **Figure 2-15**.

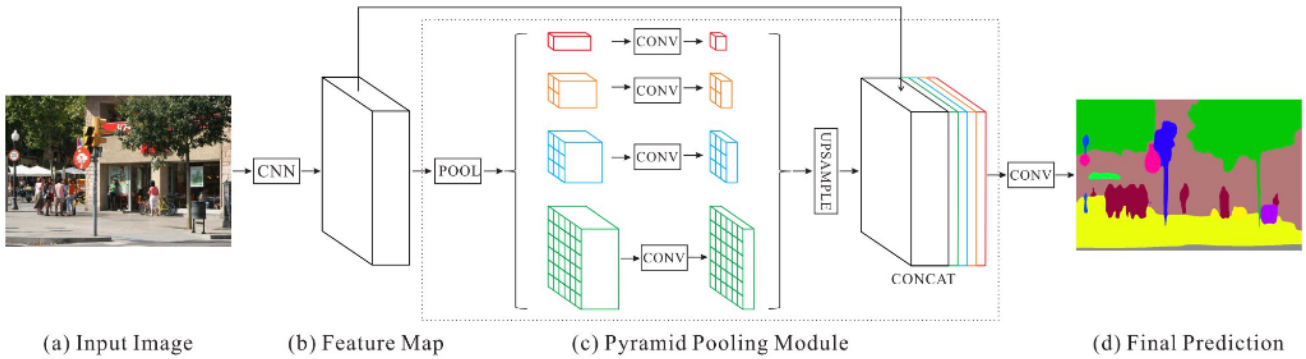


**Figure 2-15:** The structure of Pyramid Scene Parsing Network (PSPNet), from [80]. It uses a residual network (ResNet) as the backbone for feature extraction. These feature maps are then fed into a PSP module for pattern distinction in various scales.

They are then processed by $1 \times 1$ convolutional layers and concatenated together with the initial feature maps. This module could help in capturing both local and global context information in input images. Chen et al. proposed an attention mechanism that learns to softly weigh multi-scale features at each pixel location [81]. They use a state-of-the-art semantic segmentation model and train it with multi-scale images, shown in **Figure 2-16**. The attention module replaces the average and max-pooling layers and outperforms them in accuracy.

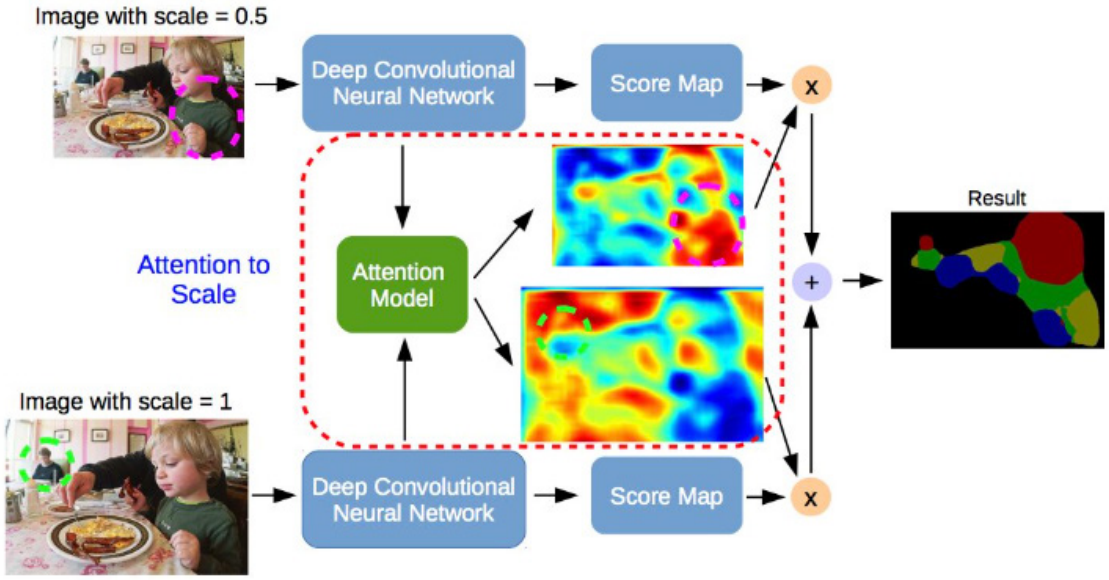


**Figure 2-16:** The structure of the Attention-based model for semantic segmentation, from [81]. The attention module here replaces the average and max-pooling layers and outperforms them in accuracy.

Besides, there are also other popular modules for semantic segmentation models, such as regional convolutional network (R-CNN), Dilated convolution models, GAN-based models.

## 2.5.2 Video Semantic Segmentation

Video semantic segmentation is one of the extended tasks for image semantic segmentation. Video object segmentation is a type of semi-supervised problem, which means that only the correct segmentation mask of the first frame of the video is provided, and then the labeled target is segmented at the pixel level in each subsequent frame. It could also be regarded as a pixel-level target tracking problem to some extent. It has many application scenarios, which make use of computers perceive and scenario understanding so that it is a crucial part of applications such as robot vision and autonomous driving.

However, compared to a single image, it is easier for us to obtain video data, and the video data itself has strong frame redundancy and uncertainty. If we send the video directly to the image segmentation frame by frame in the model, it will inevitably bring much computational overhead.

Moreover, due to changes in the moving objects in the scene will also cause instability of the segmentation results. For example, the previous frame of an object is category A. However, suddenly when it reaches the middle few frames and becomes category B, there is a phenomenon of inconsistent semantic categories inside the object. Therefore, the current main research focus of video semantic segmentation could roughly be divided into two directions: the first is how to use the timing information between video frames to improve the accuracy of image segmentation, and the second is how to use the similarity between frames to reduce the model computational costs and increases the speed and throughput of the model. Later we are going to introduce related works in those two categories.

Netwarp module was first proposed for accuracy improvement in the video semantic segmentation area in [82]. The primary function of the Netwarp module is to use optical flow to move the features of the previous frame to the current frame, and then it helps in feature enhancement. The optical flow is defined as the vector of the corresponding pixel movement between two images. This structure can be inserted between frames of the video, as shown in **Figure 2-17**.

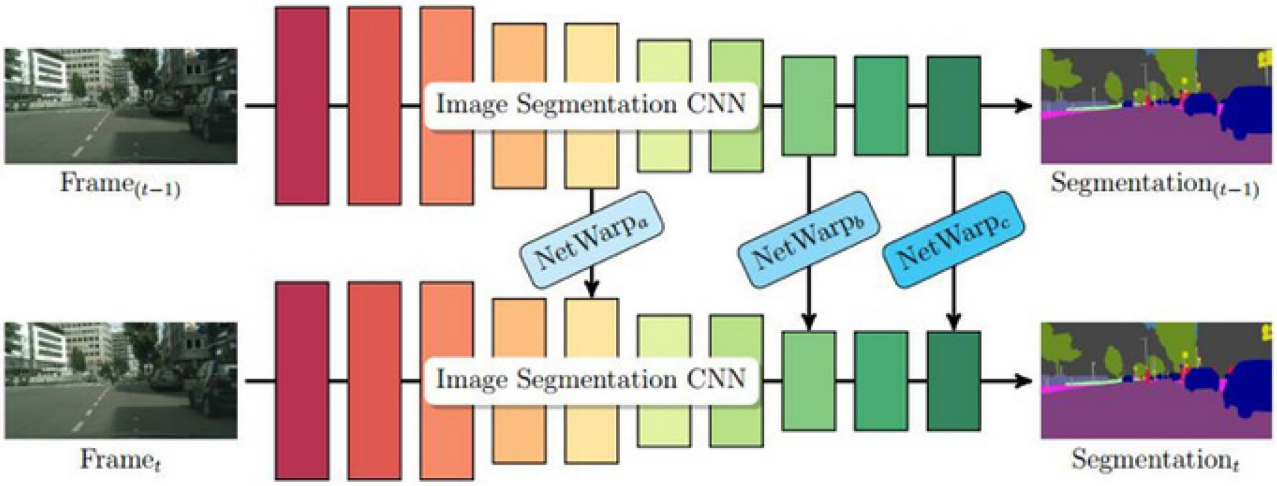


**Figure 2-17:** The structure of Netwarp, from [82]. The optical flow is defined as the vector of the corresponding pixel movement between two images. The primary function of the Netwarp module is to use optical flow to move the features of the previous frame to the current frame, and then it helps in feature enhancement.

The input of the model is two consecutive frames, $(t-1)$ represents the previous frame, and $t$ represents the current frame. The first step is to calculate the optical flow $F_t$ , the optical flow calculation is in the form of offline, that is, each optical flow is calculated in advance. Then they send the optical flow and two frames of images to a module called Transform Flow. This module is composed of a small fully convolutional network module. It is designed to supplement the optical flow information with image information, as shown in **Figure 2-18**.

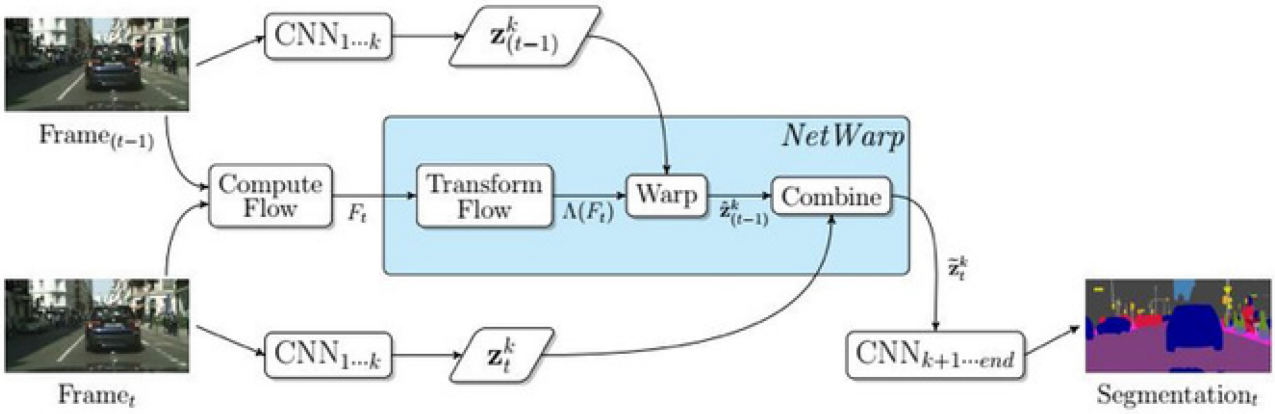


**Figure 2-18:** The structure of the entire network for semantic segmentation, from [82]. It takes two consecutive frames as input and calculate the optical flow offline. Then they send the optical flow and two frames of images to a module called Transform Flow. This module is composed of a small fully convolutional network module.

And then, they use the flow of transform to warp the feature of the previous frame to the current frame (the specific implementation of warp is to use bilinear difference operation based on the current frame find in the corresponding feature points of the previous frame from the optical flow information). Finally, the information of the current frame and previous frames' information are combined to obtain the final feature representation. The results of this network surpass PSPNet in segmentation precision.

Besides, there are also works related to it, such as the Spatio-Temporal Transformer GRU module [83]. It uses information from multiple frames to improve the precision of segmentation.

For the models that aim to reduce computational costs, Deep Feature Flow is a typical example of them. It is proposed in [84]. The article's starting point is that the difference between frames of deep features in the video is relatively small. Furthermore, the time and computational cost of acquiring deep features are extremely large for each frame (especially in some deep networks). Thus, they consider using optical flow to warp the previous features to the current frame, thereby reducing the amount of calculation. The structure of it is shown in **Figure 2-19**.

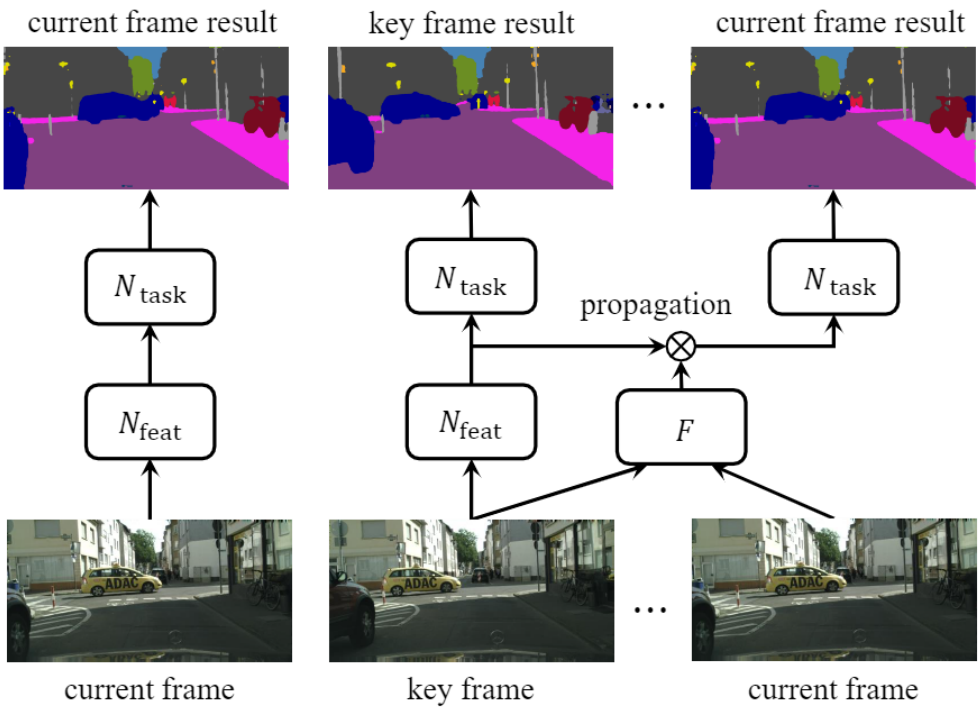


**Figure 2-19:** The structure of the entire network of Deep Feature Flow, from [84]. Here, optical flow is used to warp the previous features to the current frame, thereby reducing the amount of calculation.

First, a frame will be selected as a key frame. Here, the key frame means it would get the deep feature through the entire network. For the other frames, it will calculate the optical flow between them and the key frame by their difference. Then they use optical flow to warp the deep feature of the key frame to the current frame to obtain the segmentation result of the current frame and obtain the loss for back transmission. Here, the key frame selection is fixed, and every k frame is selected as the key frame. Since the optical flow network is relatively shallow and the amount of calculation is much smaller than that of the segmentation network, it can significantly increase the speed of segmentation.

As a pioneering work in the acceleration of video semantic segmentation, it has inspired a lot of work, such as Low-Latency Video Semantic Segmentation [85] and TD-Net [29]. We choose TD-Net as our model for Video Semantic Segmentation.

## 2.6 Video Prediction

Video prediction is defined as a self-supervised learning task. It also demonstrated potential capabilities for extracting meaningful representations of the patterns in videos. Despite the fact that Video prediction tasks would be easy for humans with additional physical knowledge, it is still highly challenging to the deep learning algorithm. Some of the factors that contribute to such complexity are occlusions, camera movement, lighting conditions, clutter, or object deformations [86]. **Figure 2-20** shows an example of video prediction tasks.

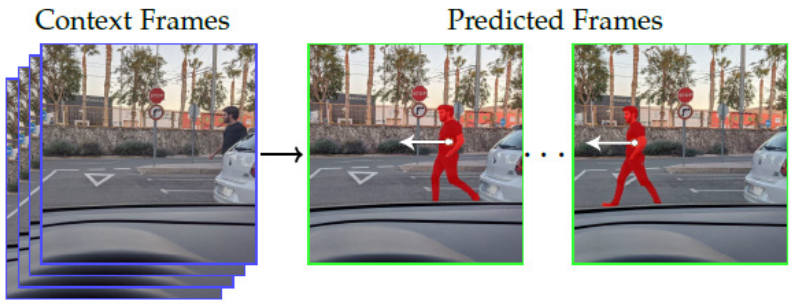


**Figure 2-20:** An example of a video prediction task, from [86]. The network would give predicted frames based on context frames.

In fact, what made the deep architectures take a leap over the traditional approaches is their ability to learn adequate representations from high-dimensional data in an end-to-end fashion without hand-engineered features [87]. Deep-learning-based models could perfectly predict diagram because it enables the extraction of meaningful Spatio-temporal correlations from video data in a self-supervised manner.

### 2.6.1 Approaches in Video Prediction

There are three types of building blocks in video prediction models: CNN, RNN, and Generative models.

Convolutional Neural Networks (CNNs) based model is a typical choice to efficiently model the spatial structure of images [88]. The short-range inter-frame dependencies are limited for convolutional operations due to their limited receptive fields, which are determined by the kernel size. Researchers have come up with ideas to circumvent it. They are: stacking more convolutional layers [59], increasing the kernel size, (3) linearly combining multiple scales [89] as in the reconstruction process of a Laplacian pyramid [90], (4) using dilated convolutions to capture long-range spatial dependencies [91], (5) enlarging the receptive fields [92, 93]. The process of a Laplacian pyramid is shown in **Figure 2-21** below.

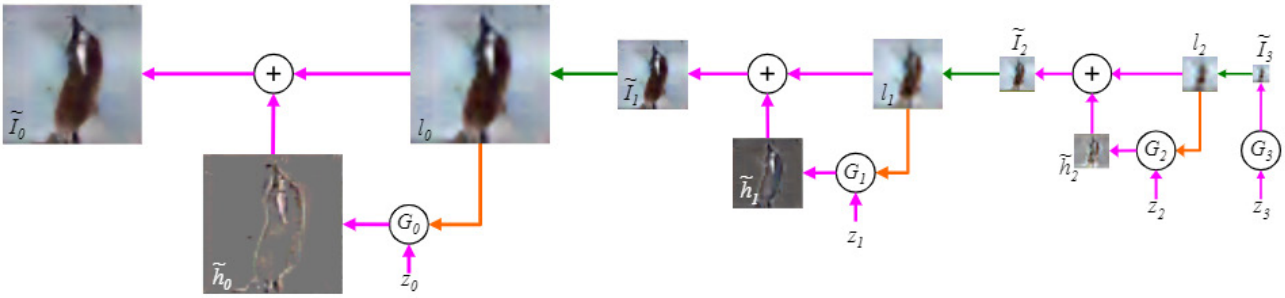


**Figure 2-21:** An illustration of the process of the Laplacian pyramid, from [90]. It takes a noise sample as input and feed it into a generative model. his process repeats across two subsequent levels to create a final sample.

It starts with a noise sample $z_3$ and uses a generative model $G_3$ to generate $\tilde{I}_3$. It is then up-sampled and used as the conditioning variable $l_2$ for the next level generative model, $G_2$. $G_2$ would

then generates a different image $\tilde{h}_3$ together with another noise sample $z_2$. After that, $\tilde{h}_3$ is added to $l_2$ to create $\tilde{I}_2$. This process repeats across two subsequent levels to create a final sample $I_0$ with full resolution.

Another popular choice of model basis for video prediction is RNN. Unlike CNN, Recurrent models were specifically designed to model a Spatio-temporal representation of sequential data such as video sequences, which makes it suitable for video prediction tasks. There are already lots of works with Vanilla RNN in video predictions [88, 94, 95, 96]. There are also excellent works with extended RNN models, such as Long Short-Term Memory (LSTM) [97] and Gated Recurrent Unit (GRU) [98]. Shi et al. introduced the LSTM-based models to the image space and proposed the ConvLSTM model in [95]. The process of encoding and inference is shown in **Figure 2-22**.

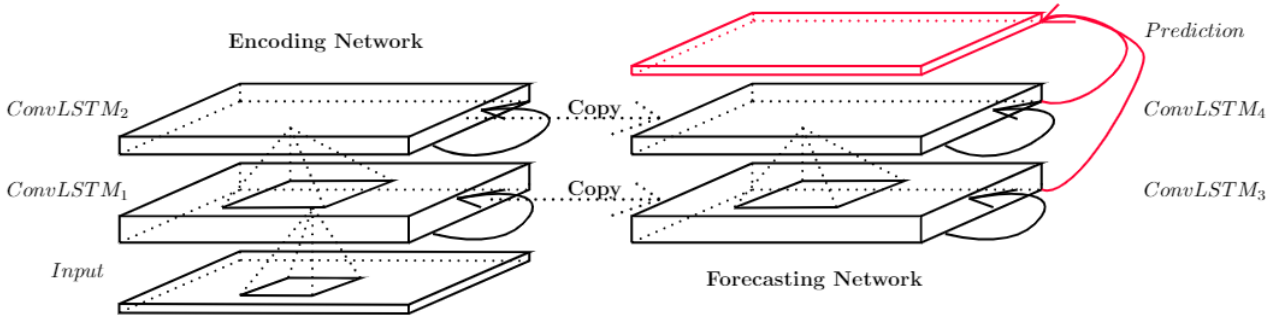


**Figure 2-22:** An illustration of ConvLSTM encoding and inference process, from [95]. It starts the encoding from input layer by layer with the ConvLSTM module. Then the encoded information will be copied to the corresponding layers in Forecasting Network. After that, a prediction would be generated.

It starts the encoding from input layer by layer with the ConvLSTM module. Then the encoded information will be copied to the corresponding layers in Forecasting Network, which has the same layer structure as Encoding Network. After that, a prediction would be generated by gathering all information from each Forecasting layer.

The third choice of model basis for video prediction is generative models. Unlike discriminative models, generative models learn the potential distribution of individual classes. In the context of video prediction, generative models are mainly used to deal with future uncertainty, as they could generate a broad spectrum of feasible predictions rather than a single outcome.

VAE is an example of them. In the video prediction context, VAEs are the basic module of many probabilistic models dealing with future uncertainty [99, 100, 101, 102]. Although these variational approaches are capable of generating various plausible outcomes, the predictions are sometimes blurrier compared to models with GANs. Additional methods like adversarial training are proposed in models with VAEs to solve this problem. The prediction process is shown in **Figure 2-23**.

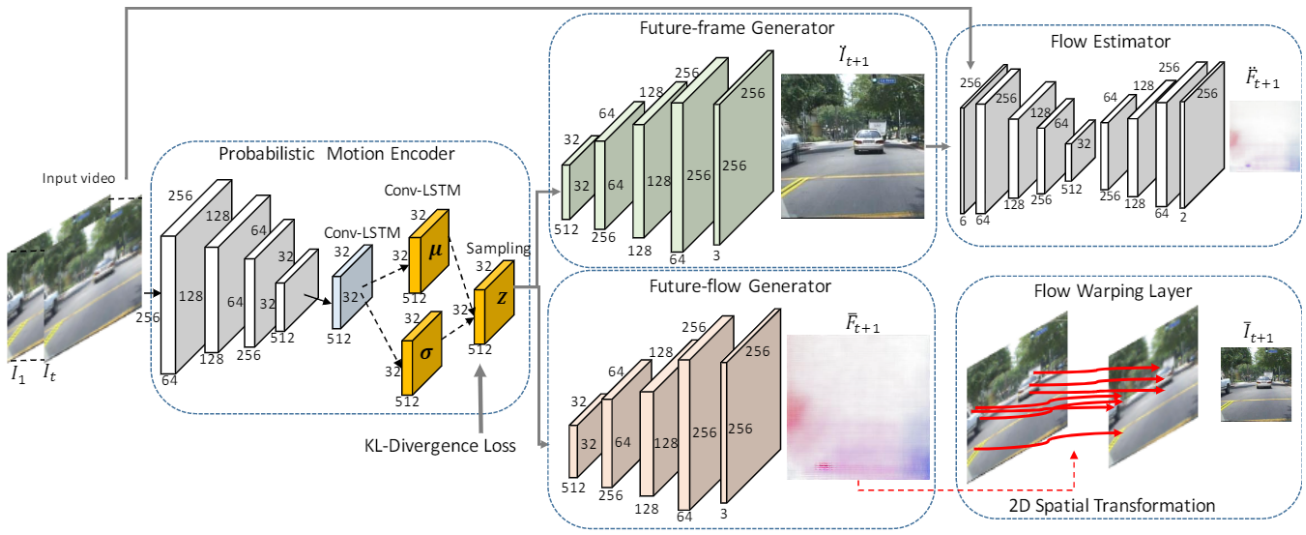


**Figure 2-23**: An illustration of the prediction process by GANs, from [101]. It starts with a probabilistic motion encoder that encodes the input frames. Then, the feature maps would be fed into Future-frame Generator and Future-flow Generator for visual frame and optical flow information. The optical flow would be fed into Flow Warping Layer for final prediction.

It starts with a probabilistic motion encoder that encodes the input frames. This encoder is also equipped with Conv-LSTM to improve the sampling process. Then, the feature maps would be fed into Future-frame Generator and Future-flow Generator, generating the visual frame and optical flow information at the same time, then the visual prediction would be fed into Flow Estimator with the original input frames to estimate another optical flow. While, at the same time, the generated optical flow would be processed in Flow Warping Layer with 2D Spatial Transformation for a final prediction of future frames.

In addition to VAE, GANs have been a more popular choice in video prediction as generative models these years, as it has been proved to be state-of-the-art in various tasks of other fields. It has already been the backbone network for many video prediction studies [89, 96, 103]. However, GANs-based models often suffer from random noise issues, such as Gaussian noise, as they are unconditioned. AMC-GAN was proposed to solve these problems. The architecture of it is shown in **Figure 2-24**.

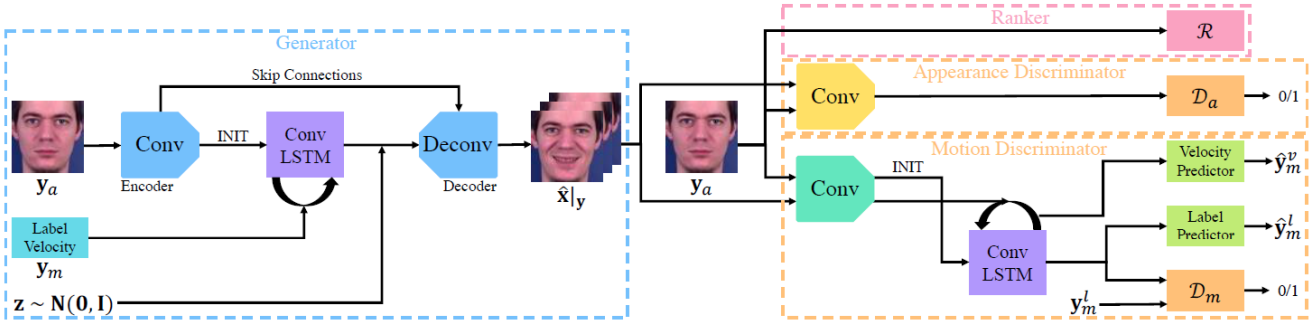


**Figure 2-24**: An illustration of AMC-GAN structure, from [104]. It consists of two parts. The first part uses an encoder-decoder structure with Conv-LSTM to generate the prediction frames with Label Velocity. The second part takes the original input frames as well as the generated predictions in the first part as input and generates rank

It consists of two parts. The first part uses an encoder-decoder structure with Conv-LSTM to generate the prediction frames with Label Velocity. The second part takes the original input frames as well as the generated predictions in the first part as input and generates rank, appearance, and motion prediction information separately by individual discriminators. It could somehow alleviate the problem of an unconditioned situation.

Compared with CNN and RNN models, generative models show a more significant potential for visually plausible and highly diverse predictions. That is also why we chose a generative model for our Video Prediction Module.

# Chapter 3. Flame and Smoke Analysis System

In order to provide analysis for flame and smoke with high accuracy, we propose an RGB image-based Flame and Smoke Analysis System for real-time analysis. A detailed structure of our system is shown in **Figure 3-1**, which is a detailed version of **Figure 1-2**.

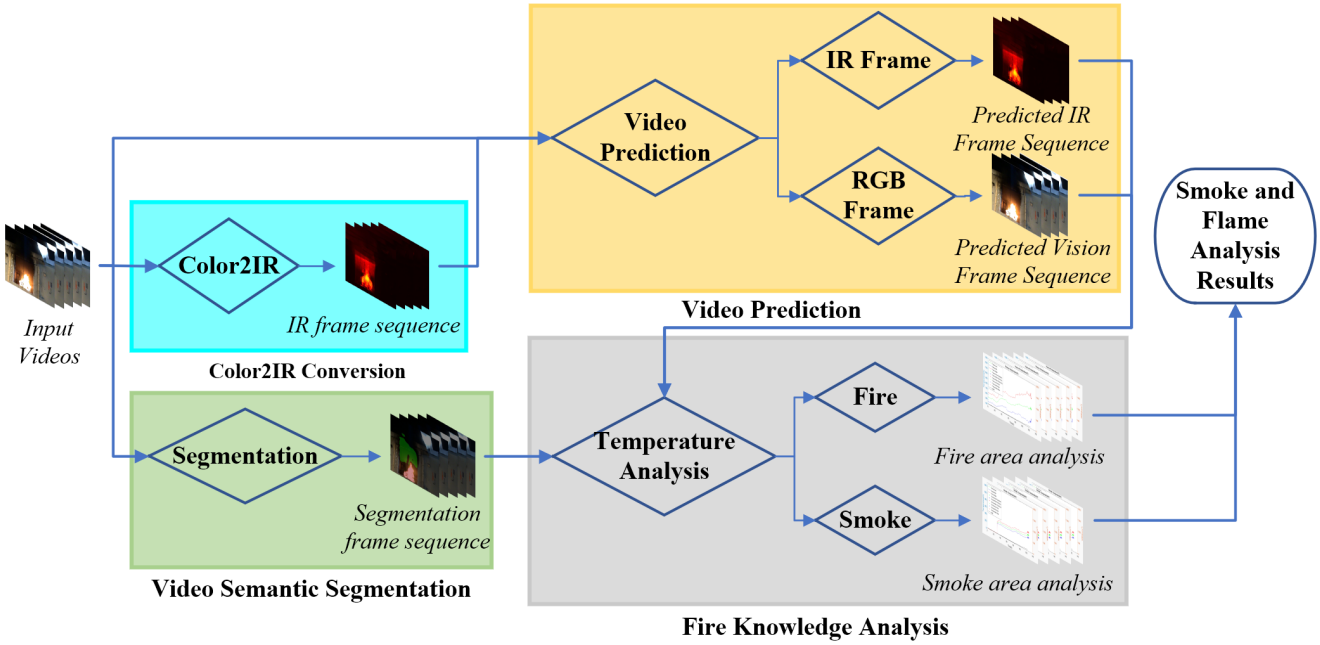


**Figure 3-1**: A detailed structure of our system. First, the Color2IR Conversion module converts RGB video frames into InfraRed (IR) frames. Next, Video Semantic Segmentation Module helps extract flame and smoke areas from the scene. After that, a Video Prediction Module takes the RGB video frames and IR frames as input and produces predictions of the subsequent frames of their scenes. Finally, a Fire Knowledge Analysis Module predicts if flashover is coming or not

It consists of 4 modules, which finish their tasks and collaborate to generate a reliable analysis result. First, the Color2IR Conversion module converts RGB video frames into InfraRed (IR) frames, providing crucial thermal information of fire. Next, Video Semantic Segmentation Module helps extract flame and smoke areas from the scene in the RGB video frames. After that, a Video Prediction Module takes the RGB video frames and IR frames as input and produces predictions of the subsequent frames of their scenes. Finally, a Fire Knowledge Analysis Module predicts if flashover is coming or not, based on fire knowledge criteria such as thermal information extracted from IR images, temperature increase rate, the flashover occurrence temperature, and increase rate of lowest temperature.

In the following sub-chapter, we will introduce the details of the four modules one by one.

## 3.1 Color2IR Conversion Module

As one of the most crucial sub-modules in our system, Color2IR Conversion aims to provide corresponding IR images that could tell temperature from a visual image captured from a standard camera that could be taken into fire rescue with firefighters.

Another reason for its importance is that it helps transform an image from the visual domain to the temperature domain. That is an excellent expansion of input information, especially for fire research, which is very sensitive to temperature information. As a matter of fact, temperature information, whether captured from a single censor as curve data that varies with time, or captured from IR cameras that tell the temperature of a specific region, has been the foundation of various fire research works from past to nowadays. Though some other parameters could provide similar or even slightly better support for fire analysis, like HRR and flame height, the temperature remains the leading one. It could be further analyzed to generate more parameters.

We choose GAN and Attention mechanism as the basis of this sub-module because of high performance on other existing image-related tasks, such as style transfer and image super-resolution. Another part of the reason is the dataset. We have extra image data from room fire tests without flashover, which could help to increase the size of the dataset to train this sub-module.

A general process of this sub-module is shown in **Figure 3-2**. The input videos would be cut into frames and processed as an individual unit. Besides, the images of input and output also represent different types of information. So, it is a kind of cross-domain image transfer task defined in the Computer Vision field.


**Figure 3-2**: An illustration of the process of Color2IR Conversion. This module takes RGB videos as input and produces IR frames sequences.

Furthermore, the resolution of visual and IR images could be different, as vision ones could easily reach over 4K resolution ($3840 \times 2160$) and top-tier IR cameras are still below HD resolution ($1280 \times 720$). Besides, as the vision and IR cameras could not be placed in the same area in tests and experiments, the view angles of the corresponding images are quite different. Thus, the image pairs become unaligned, also defined as ‘un-paired’ in Computer Vision tasks.

### 3.1.1 Architecture

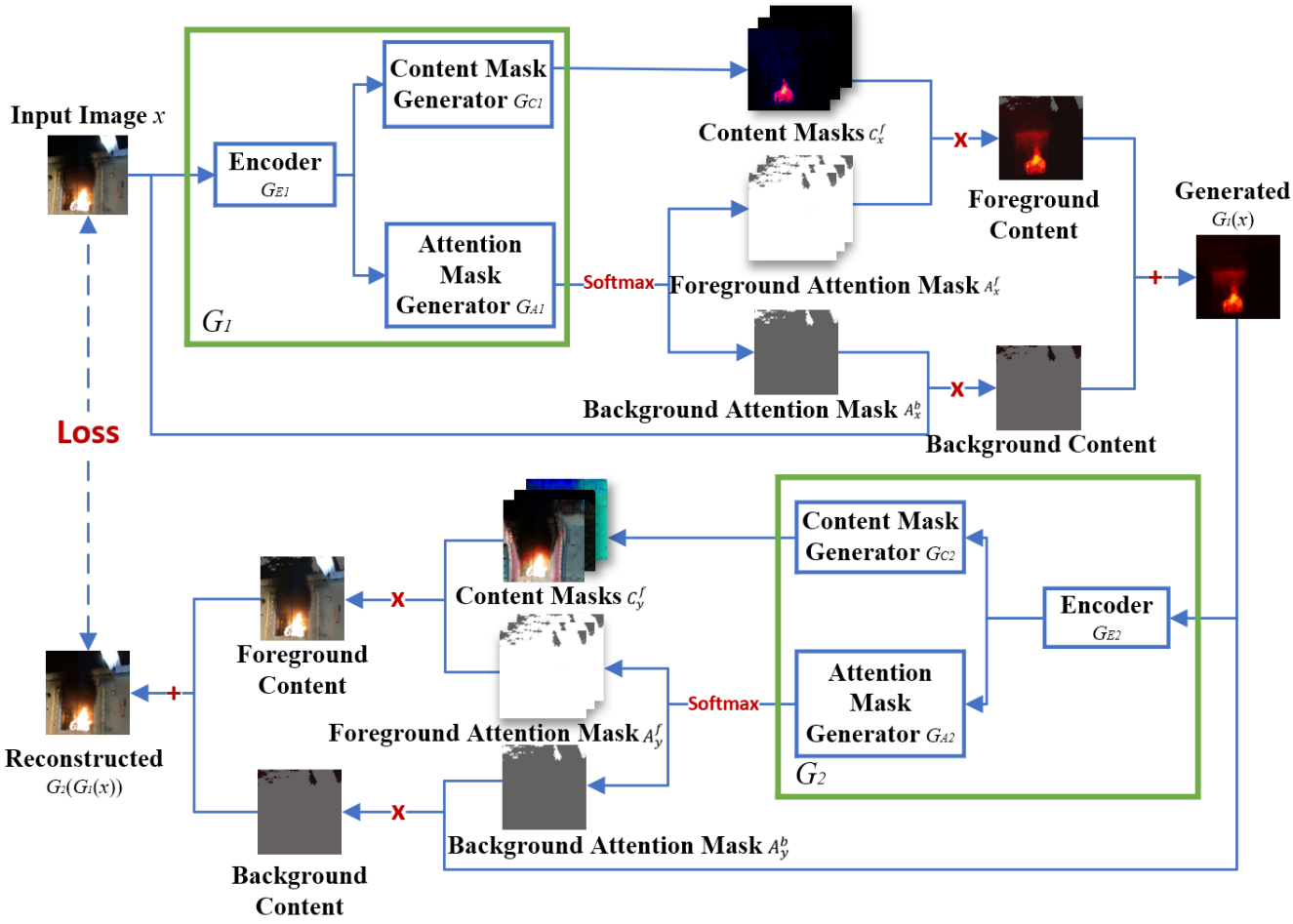


**Figure 3-3**: An illustration of DAGAN architecture. 'Red ×' stands for multiplication of matrices, 'Red +' denotes the sum of matrices, and 'Red Softmax' are the Softmax activation function. 'Red Loss' is the Cycle-Consistency Loss inspired by CycleGAN, which is only part of our Loss design for DAGAN.

In order to solve the problems listed above, we proposed a novel structure of deep neural networks: Dual-Attention GAN (DAGAN). It is optimized for the Color2IR Conversion task in our system. The architecture of DAGAN is illustrated in **Figure 3-3**. Detailed information for Loss design is illustrated in **Chapter 3.1.2**.

The overall structure of DAGAN is a loop or cycle inspired by the success of CycleGAN in un-paired image conversion. The generation and reconstruction of images use the same parts and processes while trained individually in generation and reconstruction. Besides, several types of loss functions are used in the optimization process for a better conversion result.

The input images of DAGAN are the sequences cut from the visual videos of fire scenes in the dataset, which is denoted as $x$ in **Figure 3-3**. There is no restriction for the Frame Per Second (FPS), as someone might have relatively low computational capability hardware and would like to convert them for real-time usage.

Then, the input $x$ will be fed into our generator $G_1$, which consists of an encoder $G_{E1}$ and two

mask generators: $G_{C1}$ and $G_{A1}$. $G_{E1}$ is a parameter-sharing encoder to generate low-level feature maps. While $G_{C1}$ is a content mask generator that helps in generating a set of masks $\{C_x^f\}_{f=1}^{N-1}$, that contains $(N-1)$ sets of the content feature that captured from the encoder $G_{E1}$.

Unlike $G_{C1}$, $G_{A1}$ is a generator for attention mechanism, which aims to provide attention-level feature maps from the encoded information. It is also one of the key components that enable DAGAN for excellent performance. The direct output of $G_{A1}$ is then processed by a Softmax activation function to change the scope of mapping. Then, it would produce two types of attention masks $A_x^f$ and $A_x^b$, which are classified from the range of $(0,1)$ in the output, for example, 1 represents positive, and 0 represents negative. It follows the definition of the self-attention mechanism proposed in previous works. It is simple but super effective. The foreground attention mask and background attention mask enable DAGAN to be distinct from the foreground and background of the images, which could help solve background blurry and foreground color drift. So, the two attention mask generators would produce a total number of $N$ attention masks: $[\{A_x^f\}_{f=1}^{N-1}, A_x^b]$ in the generation process. There is only one background attention mask $A_x^b$ and there is a set of $(N-1)$ foreground masks $\{A_x^f\}_{f=1}^{N-1}$. This is because the foreground is rather important than the background information. The amount of information in the foreground is also more than that of the background, as defined by the attention mask generator. This also allows the process of foreground context and background context independently.

After that, foreground information and background information extracted from input $x$ will be processed independently. For the foreground information, foreground attention masks $\{A_x^f\}_{f=1}^{N-1}$ would be used to generate the foreground content by combing the set of content masks in earlier steps. At the same time, the background attention mask would help keep a clean and tidy background of generated images by combing it with original input $x$. The final generated image $G_1(x)$ would be the sum of the two content images selected from extracted feature maps by our attention mechanism. It is calculated as the formula below.

$$G_1(x) = \sum_{f=1}^{N-1} \left(A_x^f \times C_x^f\right) + x \times A_x^b \qquad (3-1)$$

That is the end of the generation loop and also the start of the reconstruction loop. The basic idea of loop structure is that we should go back to where we start if we walk in a loop. It also works for the image conversions as the loop conversion should make the reconstruction back into the same domain as the input $x$. The reconstruction process is similar to the generation process in structure, while the training process would be independent. Let

$$y = G_1(x) \qquad (3-2)$$

And we will have an equation that describes corresponding attention masks and content masks in the reconstruction process in a way that is similar to the generation process. That equation is shown as equation 3-3 below.

$$G_2(y) = \sum_{f=1}^{N-1} \left(A_y^f \times C_y^f\right) + y \times A_y^b \quad (3-3)$$

The only difference between them is that the foreground and background areas for $x$ and $y$ would be different as they are from different domains.

In this way, we could finally form a closed-loop for the DAGAN process, starting from the input $x$ to the reconstruction of $G_2(y)$ or, in other words, $G_1(G_2(x))$ if we take the equation 3-2 into it. The process of the loop is shown below.

$$x \to G_1(x) \to G_2(G_1(x)) \approx x \quad (3-4)$$

where $x$ stands for the input image in vision domain, $G_1$ and $G_2$ are generators mentioned above.

If we take the process denoted in equation 3-3 into it, the detailed calculation process should be:

$$G_2(G_1(x)) = \sum_{f=1}^{N-1} \left(A_y^f \times C_y^f\right) + G_1(x) \times A_y^b \approx x \quad (3-5)$$

For another direction of the loop that starts from the image in the IR domain:

$$y \to G_1(y) \to G_2(G_1(y)) \approx y$$
$$G_2(G_1(y)) = \sum_{f=1}^{N-1} \left(A_y^f \times C_y^f\right) + G_2(y) \times A_y^b \approx y \quad (3-6)$$

where $y$ stands for the input image in IR domain, $G_1$ and $G_2$ are generators mentioned above.

In addition, there are two types of discriminators in DAGAN. The first type is the discriminators $D_{Y1}$ and $D_{Y2}$, They are vanilla discriminators that aim to distinguish the generated images $G_1(x)$ and real images $y$ or $G_2(y)$ and $x$.

We also proposed a brand-new type of discriminator, which is the second type of discriminator. They are $D_{YA1}$ and $D_{YA2}$. They are attention discriminators capable of taking both images and feature maps generated by the attention mask generator as input. As we have generated a total number of $N$ attention masks in the generation process. Let

$$A_x = \left[\left\{A_x^f\right\}_{f=1}^{N-1}, A_x^b\right] \quad (3-7)$$

And we make a concatenation of it with the generated images $G_1(x)$ and real images $y$. So, it should be

$$M_{1y} = [A_x, y], \quad M_{1x} = [A_x, G_1(x)] \quad (3-8)$$

Then, the attention discriminator would take $M_{1y}$ or $M_{1x}$ as input and will try to distinguish the generated images with attention masks $M_{1x}$ and the real images with attention masks $M_{1y}$.

### 3.1.2 Loss function

The loss function is also the optimization target of neural networks. Building a brand-new neural network is just half of success, even if it has excellent design. Another part of our contribution to DAGAN is the design of loss functions for it. There are several parts of loss function for DAGAN, and we are going to introduce them one by one.

There is no doubt that the first part of the loss function is an adversarial loss, the same as vanilla GAN, which is formulated as the equation below.

$$\mathcal{L}_{GAN}(G_1, D_{Y1}) = \mathbb{E}_{y\sim p_{data}(y)}[\log(D_{Y1}(y))] + \mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(1 - D_{Y1}\left(G_1(x)\right)\right)\right] \qquad (3-9)$$

In this optimization process, generator $G$ tries to minimize the adversarial loss: $\mathcal{L}_{GAN}(G_1, D_Y)$, while $D_{Y1}$ tries to maximize it at the same time. The target of $G_1$ is to generate an image $G_1(x)$ that is similar to the images from domain Y, while $D_{Y1}$ aims to distinguish between the generated images $G_1(x)$ and the real images $y$.

Similar to the relationship between equation 3-4 and equation 3-6 that lasted above, there is a similar process for the generator $G_2$ and discriminator $D_{Y2}$. Their adversarial loss is defined as the equation below.

$$\mathcal{L}_{GAN}(G_2, D_{Y2}) = \mathbb{E}_{x\sim p_{data}(x)}[\log(D_{Y2}(x))] + \mathbb{E}_{y\sim p_{data}(y)}\left[\log\left(1 - D_{Y2}\left(G_2(y)\right)\right)\right] \qquad (3-10)$$

where $D_{Y2}$ tries to distinguish between the generated image $G_2(y)$ and the real image $x$.

As a network with loop structure, there is also loop loss or cycle loss in DAGAN, as denoted in **Figure 3-2** as a red dotted line between original input $x$ and reconstruction result $G_1(G_2(x))$. The cycle-consistency loss in DAGAN is formulated as the equation below.

$$\mathcal{L}_{Cycle}(G_1, G_2) = \mathbb{E}_{x\sim p_{data}(x)}[\|G_2(G_1(x) - x)\|_1] + \mathbb{E}_{y\sim p_{data}(y)}[\|G_1(G_2(y) - y)\|_1] \qquad (3-11)$$

where the reconstruction result $G_2\left(G_1(x)\right)$ is closely related to input $x$ in pixel level and $G_1\left(G_2(y)\right)$ should match the input of $y$ under similar circumstances. Here L1 loss is used to measure the image difference in pixel level.

Besides, we also use pixel loss in DAGAN in order to constrain the generator without discriminator information at the pixel level. It could be formulated as follow.

$$\mathcal{L}_{Pixel}(G_1, G_2) = \mathbb{E}_{x\sim p_{data}(x)}[\|G_1(x) - x\|_1] + \mathbb{E}_{y\sim p_{data}(y)}[\|G_2(x) - x\|_1] \qquad (3-12)$$

Here, we also use the $L1$ loss for pixel-level measurement. It is also called identity loss in CycleGAN.

Another type of loss that we also introduce is Attention Adversarial loss in AGGAN. The original idea is similar to the formation of adversarial loss shown in equations 3-7 and 3-8. Besides, we also modified it to fit the dual-attention mechanism in DAGAN. Thus, this loss comes from the attention discriminator $D_{YA1}$ and $D_{YA2}$ and the generator $G_1$ and $G_2$. It could be formulated as the equation below.

$$\mathcal{L}_{AGAN}(G_1, D_{YA1}) = \mathbb{E}_{y\sim p_{data}(y)}\left[\log(D_{Y1}\left(M_{1y}\right))\right] + \mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(1 - D_{Y1}(M_{1x})\right)\right] \qquad (3-13)$$

where $M_{1y} = [A_x, y]$, $M_{1x} = [A_x, G_1(x)]$ as illustrated in equation 3-8.

This kind of loss could help form a stable attention mask in the training process as the attention mask generation is unsupervised, which means we have not put any annotations on the image pairs in the training set.

Besides all the losses listed above, we also propose a new loss in DAGAN to improve the stability and performance of attention masks.

We call it attention loss. Unlike the attention adversarial loss that uses the complementary information from generated images and ground truth, this is a pure attention loss that only uses the information in generated attention masks. It aims to solve the problem that attention masks could easily saturate to 1, which would ruin the foreground and background content generation. The attention loss is shown in the equation below.

$$\mathcal{L}_{At}(A_x) = \sum_{w,h=1}^{W,H} |A_x(w+1,h,c) - A_x(w,h,c)| + |A_x(w,h+1,c) - A_x(w,h,c)| \qquad (3-14)$$

where, $A_x$ is the attention mask for calculation. $W$ and $H$ is the width and height dimensions of $A_x$.

Thus, we could finally get our loss function for DAGAN optimization by piecing them all together with their independent weights. The loss function of DAGAN is formulated as follows.

$$\begin{aligned}\mathcal{L}_{DAGAN} &= \lambda_{Cycle} \times \mathcal{L}_{Cycle} + \lambda_{Pixel} \times \mathcal{L}_{Pixel} + \lambda_{At} \times \mathcal{L}_{At} + \lambda_{GAN} \times (\mathcal{L}_{AGAN} + \mathcal{L}_{GAN}) \\ &= \lambda_{Cycle} \times \mathcal{L}_{Cycle}(G_1, G_2) \\ &+\lambda_{Pixel} \times \mathcal{L}_{Pixel}(G_1, G_2) \\ +\lambda_{GAN} \times \big(\mathcal{L}_{GAN}(G_1, D_{Y1}) &+ \mathcal{L}_{GAN}(G_2, D_{Y2}) + \mathcal{L}_{AGAN}(G_1, D_{YA1}) + \mathcal{L}_{AGAN}(G_2, D_{YA2})\big) \\ &+\lambda_{At} \times \Big(\mathcal{L}_{At}(A_x) + \mathcal{L}_{At}\big(A_y\big)\Big) \end{aligned} \qquad (3-15)$$

where $\lambda_{Cycle} = 10$, $\lambda_{Pixel} = 1$, $\lambda_{GAN} = 0.5$, $\lambda_{At} = 1 \times 10^{-6}$ in our setup.

# 3.2 Video Semantic Segmentation Module

Besides the Color2IR Conversion module, another sub-module that links to the input images is Video Semantic Segmentation Module. It is used to generate semantic and area information for the input videos for fire scenes. There is no doubt that humanity is capable of recognizing fire and smoke patterns easily in most cases. At the same time, the situation for firefighters is quite different.

Unlike a house with good lighting conditions and walkways in a normal situation, firefighters usually face a super dark room with hot gases around them. That will dramatically increase the difficulty of flame and smoke recognizing. In addition, the information captured by un-trained humans is usually in a low level, which means it could be incorrect in the next few seconds as fire is developing and not useful for the setting of the rescue plan for firefighters. Though many types of fire alarm censors are already equipped in the room, they could only provide point and curve data, which are simple variations like temperature. Plus, we could not figure out the exact location for them. As a result, those censors could only be used for early alarm in the house or department instead of developing fire scenes.

Segmentation is one of the traditional image processing tasks for deep neural networks. Different structures of neural networks have been proposed with high accuracy and speed. So, they are also a good choice for our smoke and flame segmentation tasks.

As a matter of fact, processing of flame and smoke from videos usually requires providing real-time video semantic segmentation results for input videos. It requires both accuracy in segmentation and speed in processing. TD-Net [29] is a type of neural network for video semantic segmentation that is made of efficient networks, which are smaller in network structure and number of parameters. While, at the same time, it could provide segmentation results as accurate as those 'heavy' networks. This is also the reason why we chose TD-Net for our Video Semantic Segmentation Module.

The idea of TD-Net is inspired by Group Convolution, which shows that extracting features with separated filter groups instead of only one will allow for a better model parallelization and help learn better representations. In other words, it indicated that features extracted from a particular high-level layer of a very deep CNN could be approximated by composing features extracted from several shallower sub-networks. The sub-networks design and Attention Propagation Module (APM) are the core of TD-Net, contributing to the fast and consistent segmentation results. An illustration of the detailed structure of TD-Net is shown in **Figure 3-4**.

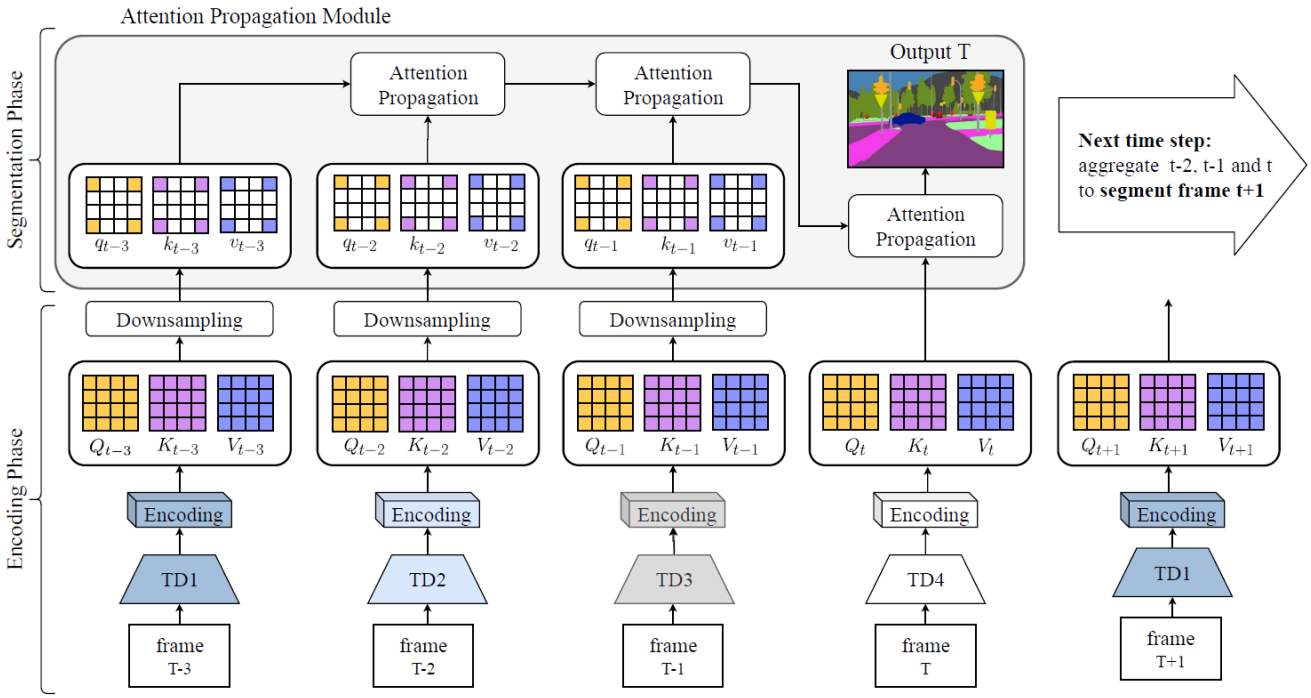


**Figure 3-4**: An illustration of the detailed structure of TD-Net, from [29]. It takes 4 frames as input and gives segmentation results based on them. TD-Net takes advantage of the information between frames for a better segmentation result.

There are two phases in this process. The first one is the Encoding Phase. This part is used to generate feature maps that are further used in the later phase. To be specific, it extracts Value feature maps which are path-specific, and Query and Key maps for across-frames correlations between pixels. They calculate the attention from Value, Query, and Key as a self-attention mechanism formulated

as the equation below.

$$\boldsymbol{Aff}_p = Softmax\left(\frac{Q_t K_p^T}{\sqrt{d_k}}\right) \quad (3-16)$$

where $d_k$ is the dimension of $Q_t$ and $K_p$.

Then, they merged those feature maps together at current frames, and previous $(m-1)$ frames as follow:

$$V_t' = V_t + \sum_{p=t-m+1}^{t-1} \phi\left(\boldsymbol{Aff}_p V_p\right) \quad (3-17)$$

With this self-attention mechanism, they could capture the non-local correlation between pixels across frames effectively. Then they also use downsampling to reduce the computation costs for it.

Next, in the segmentation phase, they propose a propagation approach that measures the attention of $m$ neighboring frames. It is formulated as the following.

$$v_p' = \phi\left(Softmax\left(\frac{q_t k_p^T}{\sqrt{d_k}}\right) v_{p-1}'\right) + v_p \quad (3-18)$$

where $q_t$, $k_p$ and $v_p$ is the downsampled version of $Q_t$, $K_p$ and $V_p$.

Then the final feature representative at frame $t$ is computed as:

$$V_t' = \phi\left(Softmax\left(\frac{Q_t k_{t-1}^T}{\sqrt{d_k}}\right) v_{t-1}'\right) + V_t \quad (3-19)$$

And the segmentation maps are generated by the equation as follows.

$$S_m = \pi_m(V_m') \quad (3-20)$$

where $\pi_m$ is the final prediction layer of sub-networks $m$.

Besides, they also use a Grouped Knowledge Distillation mechanism to enhance the sub-feature maps in the full feature space. The loss function is illustrated in equation 3-21.

$$\mathcal{L} = CE\left(\pi_S(V_i', gt)\right) + \alpha \cdot KL(\pi_S(V_i') || \pi_T(\sum f)) + \beta \cdot KL(\pi_S(V_i) || \pi_T(f_i)) \quad (3-21)$$

where $CE$ denotes $CrossEntrophy$ loss, $KL$ is the KL-divergence. $\pi_S$ is the prediction of student network and $\pi_T$ is that of teacher network.

In our system, we set the $m$, which is the number of sub-networks to 2, in order to balance the speed and accuracy of it. Thus, it is a TD$^2$-PSP50 model with a PSPNet-50 as backbone and PSPNet-101 as teacher network for knowledge distillation.

Furthermore, we also apply data augmentation for this module by generating synthetic images. An example of it is shown in **Figure 3-5**.

**Figure 3-5**: The generation of a synthetic image for data augmentation. The 2 images at the left are the source of the scene without fire and flame patterns. The flame pattern is superimposed to the scene for synthetic image generation.

The synthetic flame patterns are generated in Blender, a free and open-source 3D computer graphics software for computer animation. We use it to build and generate life-like flame patterns and merge them into real scenes where the fire did not happen. Besides, in order to make those images more life-like, we use $\alpha$-channel edge processing for blending, which is an excellent technique to fuse the foreground and background in edges.

## 3.3 Video Prediction Module

Video Prediction Module is one of the sub-modules in our system that is directly related to prediction purposes. As a matter of fact, we aim to use the power of neural networks to provide reliable visual results for fire scenes.

In the traditional fire safety research area, the simulation engine is the most used method for the prediction of fire development. However, they are usually complicated as there are a huge number of environment settings. Besides, the prediction speed of simulations is relatively low. In order to solve those problems, we turn to a prediction algorithm by deep neural networks for help. They have higher prediction speed and have similar prediction quality compared with the results of simulation engines.

Generative models are the state-of-the-art methods in this field nowadays for the various studies done on video prediction tasks in the previous years, as we summarized in previous chapters. Encoder-Decoder models could give predictions with diversity, and GANs models are capable of giving naturalistic predictions. A combination of them will promise a prediction with stochasticity and plausibility. Stochastic Adversarial Video Prediction (SAVP) is a model of this type, and this is also the main reason for us to choose it in our Video Prediction Module. An illustration of the detailed structure of the SAVP model is shown in **Figure 3-6**.

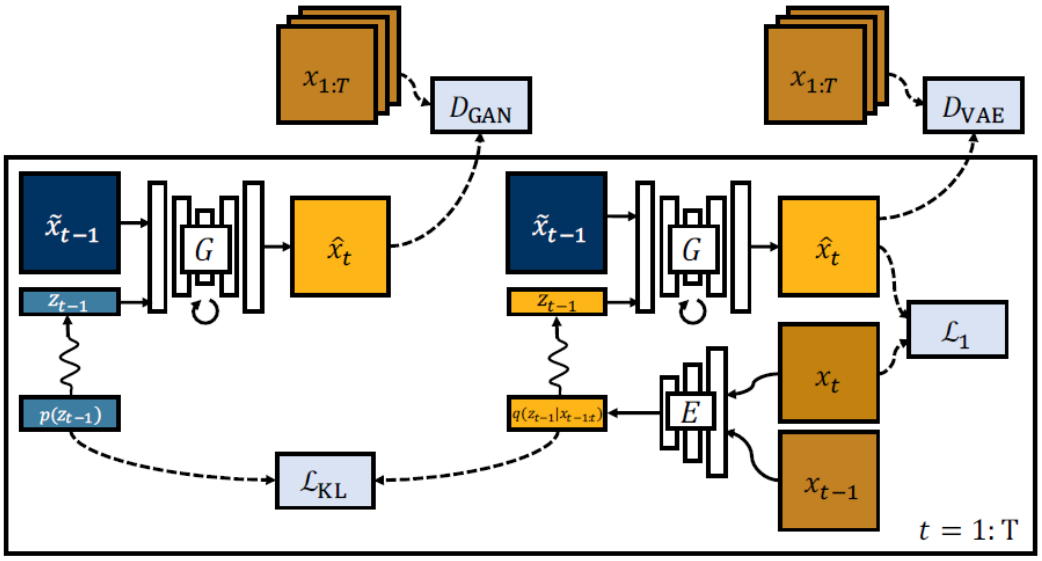


**Figure 3-6:** An illustration of the detailed structure of SAVP, from [100]. The variable in the deep blue rectangle is the input of this network. It will be processed independently by 2 independent generators in GAN and VAE. The results of VAE and GAN are linked by a KL divergence loss.

Their model consists of two parts.

The first part is made of a VAE as well as a generator. The generator $G$ predicts the future frames give the previous ones $\tilde{x}_{t-1}$ and latent codes $z_{t-1}$, thus it specifies a distribution $p(x_t|x_{0:t-1}, z_{0:t-1})$, which is a fixed variance Laplacian distribution with mean as $\hat{x}_t = G(x_0, z_{0:t-1})$. For the VAE part, they use a conditional VAE which has a conditional encoder and decoder on the previous frames $\hat{x}_t$ or $x_t$. Then, they rewrite the reconstruction term to allow the backpropagation through the encoder. That term is formulated as follows.

$$\mathcal{L}_1(G, E) = \mathbb{E}_{x_{0:T}, z_t \sim E(x_{t:t+1})|_{t=0}^{T-1}} \left[ \sum_{t=1}^{T} \left\| x_t - G\left(x_0, z_{0,t-1}\right) \right\|_1 \right] \qquad (3-22)$$

where $\hat{x}_t$ denotes reconstructed frames, $x_t$ is the ground truth frames. $z_t$ is the latent variables.

Besides, they also use a regularization term for the encoder to approach the prior distribution with the posterior one. It is shown as follows.

$$\mathcal{L}_{KL}(E) = \mathbb{E}_{x_{0:T}} \left[ \sum_{t=1}^{T} \mathcal{D}_{KL}(E(x_{t-1:t}) || p(z_{t-1})) \right] \qquad (3-23)$$

where *KL* is the KL-divergence.

So, the optimization of VAE involves minimizing the objects listed above in equation 3-22 and equation 3-23. They also use parameters for weighting, shown in the equation below.

$$G^* E^* = \arg\min_{G,E} \lambda_1 \mathcal{L}_1(G, E) + \lambda_{KL} \mathcal{L}_{KL}(E) \qquad (3-24)$$

For the second part, GAN, which is shown in the left part of **Figure 3-4**, generator $G$ aims to provide a prediction of future frames $\hat{x}_{1:T}$. While the discriminator $D$ would try to distinguish the generated frames $\hat{x}_{1:T}$ with the real ones $x_{1:T}$. Thus, the generator would be trained using binary cross-entropy loss, formulated as follow.

$$\mathcal{L}_{GAN}(G,D) = \mathbb{E}_{x_{1:T}}[\log D(x_{0:T-1})] + \mathbb{E}_{x_{1:T},z_t\sim p(z_t)|_{t=0}^{T-1}}\left[\log\left(1 - D\big(G(x_0, z_{0:T-1})\big)\right)\right] \quad (3-25)$$

For the generator, it could be learned with an adversarial process, formulated as follow.

$$G^* = arg\min_G \max_D \mathcal{L}_{GAN}(G,D) \quad (3-26)$$

Finally, we could get the optimization objective with VAE and GAN part together, shown in the equation below.

$$G^*, E^* = arg\min_{G,E}\max_{D,D^{VAE}} \lambda_1\mathcal{L}_1(G,E) + \lambda_{KL}\mathcal{L}_{KL}(E) + \mathcal{L}_{GAN}(G,D) + \mathcal{L}_{GAN}^{VAE}(G,E,D^{VAE}) \quad (3-27)$$

For the detail of the structure, they use Conv-LSTM in generators $G$ and the discriminator in SNGAN as discriminator $D$.

# 3.4 Fire Knowledge Analysis Module

Though Deep Learning has been proved in various fields, various areas have developed in the past few decades continuously without it. As a matter of fact, there are already a variety of 'traditional' methods, such as mathematical models or experienced-based models.

Fire research is a typical example of it, which has a significant contribution to fire safety and research purpose. For example, one of the most crucial information sources is sensors. Censors could consistently capture the information from a specific location. The most-used sensors in fire safety research are the temperature sensors, which could capture the temperature information as point data or curve data the flows with time. With the related information from several sensors, researchers could build a graph representing some fire development features in a specific area. In addition, traditional classification models such as regression models could be built to give binary results for the fire development. Although a variety of output is not comparable to modern deep learning models, as the number of parameters for those traditional ones is exponentially less than deep learning ones, it steadily promoted fire safety research in the past.

Another example of it is the simulation method, which is widely used in testing and validation purposes for fire safety research. Compared with deep learning methods that explicitly define the real-world physical restriction for fire development, the simulation engines require a clear definition of all the related variables for the fire development, such as ventilation, material, and dimension. With all the variables properly set, simulations could help reproduce the test or validation setup. However, it also brings another problem: complexity. As real-world situations are often much more complicated than the setup in simulation engines, even a simulation test with hundreds of well-defined variables is only a simple one compared with houses and department rooms, not to mention the situations firefighters face.

Inspired by the idea listed above, we propose the Fire Knowledge Analysis Module. Here we

choose the graph analysis and statistical analysis methods in the first part that we mentioned above, as the results from the simple simulation conditions might not be applicable to the real ones that firefighters face. The details of the criteria that we used in this module are shown in **Table 3-1**.

**Table 3-1:** Criteria used in our Fire Knowledge Analysis Module.

| Criteria Base | A detailed description of Criteria |
|---|---|
| Smoke | The average temperature of the smoke layer is above 600°C |
| | The minimum temperature of the smoke layer is above 450°C |
| Flame | The increase rate of the temperature of the flame is above 15°C per second |

In the analysis and prediction of temperature variation for the future, there are 3 criteria that we use. When all the criteria are met, a flashover happens. The calculation of those numbers for the average temperature of the smoke layer, the Minimum temperature of the smoke layer, and the increase rate of the temperature of the flame is done by statistical information analysis. It could be regarded as a tangent of specific points representing the original input, shown below.

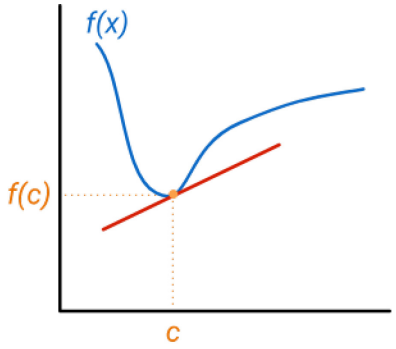


**Figure 3-7**: An illustration of applicable predictions and the temperature data curve. The blue curve is the temperature data curve, and the $c$ is a time point that we want to analyze. $f(c)$ is the temperature value at time $c$. The red line is the tangent line of the point $c$.

Just as the definition of derivatives, we could approximate the future point with current data and the tangent, formulated as the equation below.

$$f(c_f) = f(c) + \sigma \cdot (c_f - c) \qquad (3-28)$$

where $c$ is the point of original frames, $c_f$ denotes the points for the future, $\sigma$ is the tangent value.

Since the time domain of the statistical temperature graph is discrete, we need to link them together to form a continuous curve. Though the curve generated is not first-order continuous in the time domain, there is no need to worry about it as we are not going to calculate the derivatives from the curve.

Besides, the tangent is calculated independently for each point on the temperature graph, which means that we could have a piece of tangent information updated with every input frame. It could also help in preventing a collapse in fluctuation, shown in **Figure 3-8**.

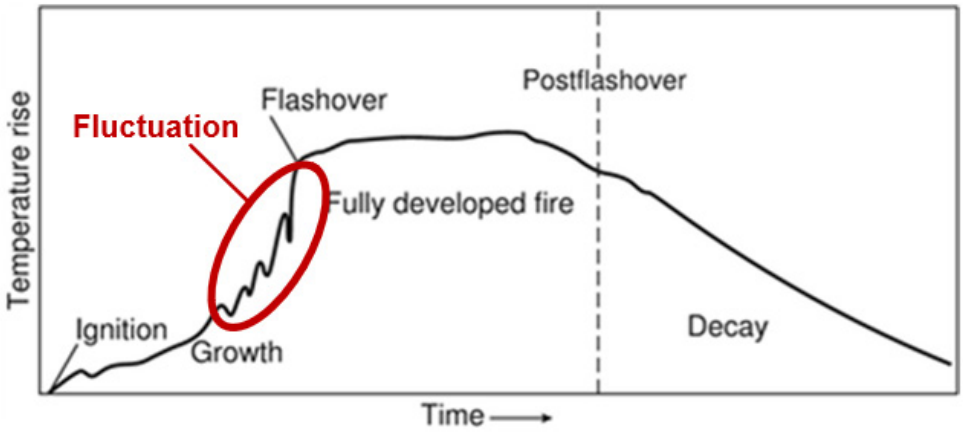


**Figure 3-8**: An illustration of temperature variation with time. The fluctuation is between flashover and the growth stage of fire development, marked in the red circle.

Our Fire Knowledge Analysis Module set the FPS of input frames to 1, which is the same as the settings in previous modules, to reduce computation cost and remain precision analysis.

# Chapter 4. Evaluation

We evaluated the performance of our system as well as the sub-modules on independent tasks. For the performance of each sub-modules, we not only introduced evaluation metrics and then evaluated the performance on it with the existing dataset for specific tasks that were introduced in the chapter before but also compared their results with other state-of-the-art approaches quantitatively and qualitatively. For the DAGAN proposed by our work, we also finished an ablation study to explore the influence of the dual-attention mechanism and the design of loss function for optimization besides the steps listed above.

Finally, we introduce the evaluation matrices for action prediction similar to the cases in the prediction of fire phenomenon, then evaluate the whole system's performance in flashover prediction and compare it with other flashover prediction models.

## 4.1 Dataset Preparation

### 4.1.1 Dataset for sub-modules

#### 4.1.1.1 Color2IR Conversion Module

For the Color2IR Conversion Module, we tested on two datasets.

The first one is a Color2IR dataset proposed by us. It is a dataset of RGB images and IR images collected from fire safety tests and experiments. Unfortunately, as those images in the dataset are confidential, we could not share all the information about the Color2IR dataset. However, we have done all the training and testing on the dataset, and our Color2IR module works perfectly on it.

Here are some of the details that are allowed to be released. The Color2IR dataset contains around 1800 image pairs from 17 fire safety experiments. They are burning tests for single and multiple items that monitor the fire development in a room of modern houses under different circumstances. The visual images are cut from the video recording sources, and IR images are collected from the same source. All the image pairs have been verified and synchronized, and a description of them is shown in **Table 4-1** below. Our IR images use a 1024-level constant colorbar that ranges from $(280, 1400)$ *Kelvin (K)* to transform the temperature information to the color domain.

**Table 4-1:** Statistical numbers of Color2IR Dataset, including the number of samples with single/multiple burning items.

| Partitions | Number of samples with single burning item | Number of samples with multiple burning items | Total number of samples |
|---|---|---|---|
| Train | 755 | 721 | 1476 |
| Test | 84 | 81 | 165 |
| Validate | 82 | 82 | 164 |

Due to the reason described above, we also evaluate our Color2IR Module on a public dataset for image conversion tasks, Map2Aerial datasets. The two domains of the dataset are map and aerial in navigation. All the images are cropped from Google Maps and resized to $600 \times 600$ resolution for normalization. Detailed description for Map2Aerial dataset is listed in **Table 4-2** below. An overview of the Map2Aerial dataset can be found in **Figure 4-1** below.

**Table 4-2:** Statistical numbers of Map2Aerial Dataset, including the number of samples with buildings, vegetation, and water bodies.

| Partitions | Number of samples mainly contains buildings | Number of samples mainly contains vegetation and water bodies | Total number of samples |
|---|---|---|---|
| Train | 684 | 412 | 1096 |
| Test | 711 | 387 | 1098 |
| Validate | 703 | 395 | 1098 |

**Figure 4-1**: Samples from Map2Aerial dataset. Each pair of images contains an aerial (at left) and map (at right). The blue arrow indicates the direction of image conversion, that is, from aerial image to map image.

As shown in **Figure 4-1**, the four pairs of images in the first row are the samples that mainly contain buildings, we can see the roads, streets, and highways clearly. However, the other four pairs of images are samples that mainly contain vegetation and water bodies which are shown as the green or blue area in aerial samples. The purpose of distinguishing those two types is that many image conversion neural networks might perform well on the first type and fail to precisely convert vegetation and water bodies area.

We test DAGAN on the Map2Aerial dataset for two purposes. The first one is that DAGAN is

a novel approach proposed by us. One of the best ways to verify its performance is to compare it with other state-of-the-art methods on the traditional image conversion dataset, such as Map2Aerial. Another reason is that the transformation from the map to the aerial domain is somehow visually similar to that between the visual and IR domains. They are all about transformation tasks that extract hidden information from one domain to another.

As the Map2Aerial dataset is the only one that we are only able to provide all the training and testing results and details, we will provide evaluation results for our Color2IR Conversion Module only on it in the following part of this chapter.

#### 4.1.1.2 Video Semantic Segmentation Module

For the Video Semantic Segmentation Module, we also built a new dataset Fire Safety (FS) Segmentation dataset. It contains 40 image sequences collected from videos captured by firefighters' equipment, NRC fire tests, fire rescue videos on YouTube, and synthetic fire images. Each sequence in the dataset contains images in 2 seconds of the original videos. The number of images in the sequence depends on the FPS of its original video. The annotations of flame and the smoke area are discussed with a group of 5 people. The ratio for training and testing is 9:1. A description of the FS Segmentation dataset is listed in **Table 4-3** below. An overview of samples with their annotations from the FS Segmentation dataset is shown in **Figure 4-2** below.

**Table 4-3**: The number of images in the FS Segmentation dataset. It has 12 sources of video sequences and about 1600 images.

| Source and sequence name | Numbers of images in each sequence | FPS of the original video | Number of sequences | Total number of samples |
|---|---|---|---|---|
| Firefighters, Firerescue-1 | 48 | 24 | 2 | 96 |
| Firefighters, Firerescue-2 | 48 | 24 | 3 | 144 |
| NRC, RBF-07 | 60 | 30 | 3 | 180 |
| NRC, RBF-12 | 60 | 30 | 4 | 240 |
| NRC, M-1 | 60 | 30 | 4 | 240 |
| NRC, 16-SI-16 | 60 | 30 | 4 | 180 |
| NRC, 26-SI-26 | 60 | 30 | 4 | 180 |
| NRC, 23-SI-76 | 60 | 30 | 4 | 240 |
| YouTube, NISTvideo-1 | 48 | 24 | 4 | 192 |
| YouTube, NISTvideo-2 | 48 | 24 | 4 | 192 |
| Synthetic, Blender-1 | 60 | 30 | 2 | 120 |
| Synthetic, Blender-2 | 60 | 30 | 2 | 120 |


**Figure 4-2**: Samples and their annotations (red: flame, green: smoke) from the FS Segmentation dataset. The blue arrow indicates the direction of segmentation.

Besides the image sequence that is processed from the 'real' scenes, we also add synthetic fire and smoke image sequences, shown in the second's row of **Figure 4-2**. The synthetic smoke and flame pattern is generated in Blender, a free and open-source 3D computer graphics software for computer animation. We use it to build and generate life-like smoke and fire patterns and merge them into real scenes where the fire did not happen. We use $\alpha$-channel edge processing for blending.

#### 4.1.1.3 Video Prediction Module

For the dataset of Video Prediction, we also built a new dataset for video prediction in fire scenes as none of the current video prediction datasets are mainly about scenes with fire and smoke. We call it the (Fire Video Prediction) FVP dataset. FVP dataset consists of 60 image sequences collected from videos captured firefighters' equipment, NRC fire safety tests, and the fire rescue video of YouTube. Each sequence in the dataset contains images in 20 seconds of the original videos. The number of images in the sequence depends on the FPS of its original video. The ratio of training and testing part is 9:1. A description of the FVP dataset is listed in **Table 4-4** below.

**Table 4-4**: The number of images in the Fire Video Prediction (FVP) dataset. It has 12 sources of video and about 40000 frames.

| Name of dataset | Source and sequence name | Numbers of images in each sequence | FPS of the original video | Number of sequences | Total number of samples |
|---|---|---|---|---|---|
| FVP | Firefighters, Firerescue-1 | 480 | 24 | 4 | 1920 |
| | Firefighters, Firerescue-2 | 480 | 24 | 4 | 1920 |
| | NRC, PRF-07 | 600 | 30 | 4 | 2400 |
| | NRC, PRF-12 | 600 | 30 | 6 | 3600 |
| | NRC, M-1 | 600 | 30 | 6 | 3600 |
| | NRC, 16-SI-16 | 600 | 30 | 6 | 3600 |
| | NRC, 26-SI-26 | 600 | 30 | 6 | 3600 |
| | NRC, 23-SI-76 | 600 | 30 | 6 | 3600 |
| | YouTube, NISTvideo-1 | 480 | 24 | 6 | 2880 |
| | YouTube, NISTvideo-2 | 480 | 24 | 6 | 2880 |
| | Synthetic, Blender-1 | 600 | 30 | 3 | 1800 |
| | Synthetic, Blender-2 | 600 | 30 | 3 | 1800 |

### 4.1.2 Dataset for the entire system for flashover prediction

To test the performance of our system for flashover prediction, we built a Flashover Prediction (FP) dataset that contains the videos recorded for NRC [105] and NIST [106] fire safety tests and their analysis for flashover time by calculating the HRR and temperature information captured by sensors in the tests. Our FP dataset consists of 8 individual fire scenes. As the system does not need to train on those samples, all of them are for testing purposes. A description of the FP dataset is listed in **Table 4-5** below.

**Table 4-5**: The sequence length and flashover time of the FP dataset. It has 8 sources of videos with fire flashovers.

| Source of video | Sequence name | Sequence length (s) | Flashover time (s) |
|---|---|---|---|
| NRC | PRF-07 | 250 | 185 |
| | PRF-12 | 150 | 94 |
| | 08-SI-04 | 250 | 169 |
| | 14-SI-06 | 250 | 157 |
| | 21-SI-10 | 150 | 95 |
| | 23-SI-76 | 200 | 113 |
| | 31-SI-13 | 300 | 227 |
| NIST | NISTtest-1 | 50 | 22 |

For the entire system, it is trained in sub-module-level instead of system-level. Thus, the data listed in the table above are only for testing purposes. The training of each sub-module in the system requires extra data other than those for whole system testing.

## 4.2 Evaluation of Sub-modules

### 4.2.1 Color2IR Module

In the evaluation of the Color2IR Module, a novel model proposed by our work, we will first introduce the test results on 2 datasets introduced in the previous chapter and compare with other state-of-the-art methods for image conversion tasks performed on both datasets. Finally, we conduct an ablation study on DAGAN to verify the effect of the dual-attention mechanism and the brand-new loss function design in it.

However, as one of the datasets with IR images used for experiments on the Colo2IR Module is confidential, as explained in Chapter 4.1.1.1. Thus, in the following part, we only introduce the results on the other dataset: Map2Aerial Dataset.

#### 4.2.1.1 Experimental results with Map2Aerial dataset

For the performance of the Color2IR Conversion Module on the Map2Aerial dataset, **Figure 4-3** shows some of the visual samples generated by DAGAN, CycleGAN, pix2pix methods by cGAN, and AGGAN. The labels above the images show the source of one column of images, either Input, Ground Truth (GT), or generated by deep neural networks.

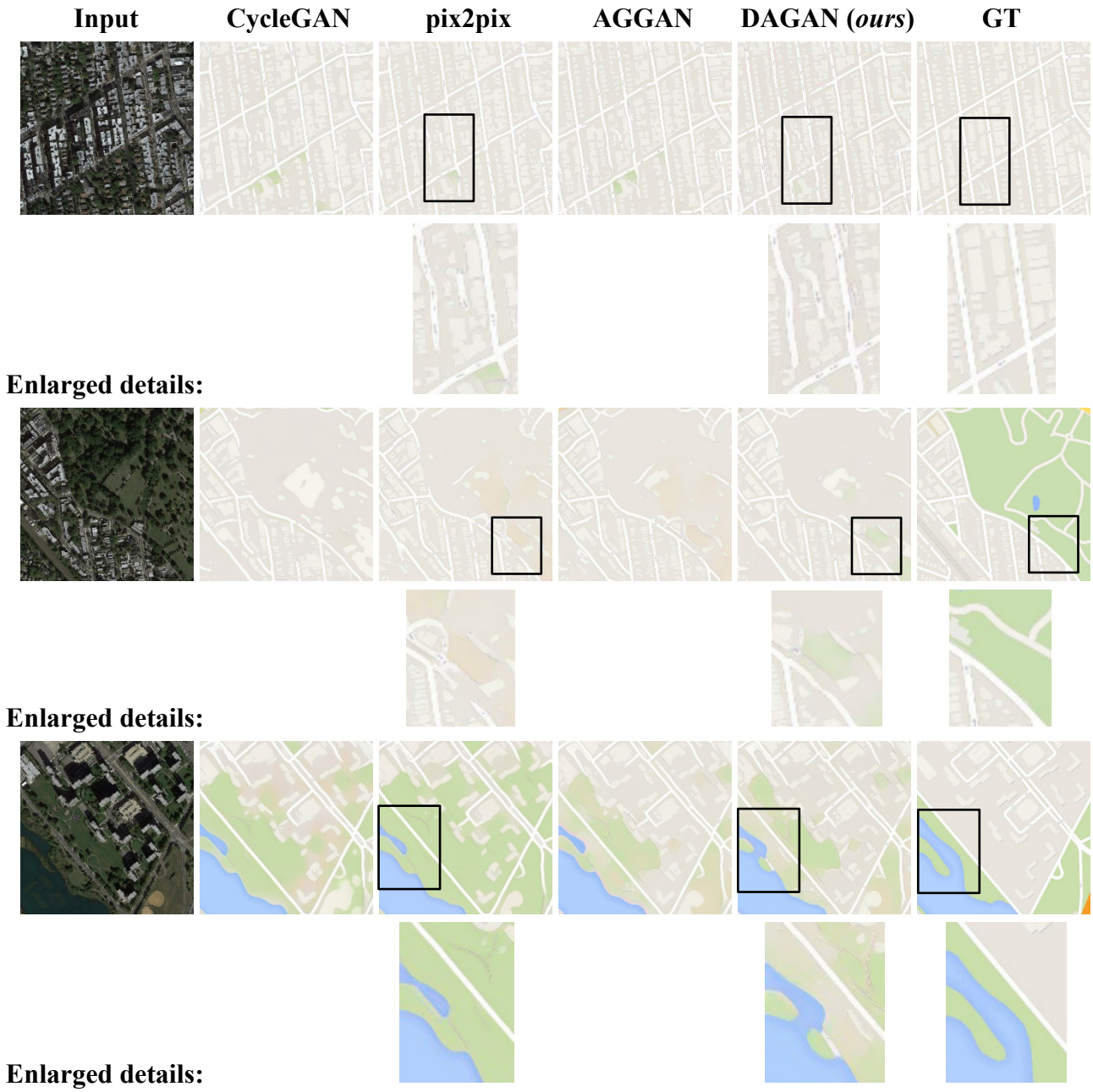


**Figure 4-3**: Images of conversion results, the label above denotes the source of each column. The row starts in 'enlarged details is the enlarged version of the rectangle area in the previous row.

From the samples in **Figure 4-3**, we could observe that the quality of generated images by DAGAN is consistently better than the CycleGAN model no matter in samples that mainly contain buildings or samples that mainly contains vegetation and water bodies. The visual result of roads, streets, vegetation, and water bodies are better in images converted by DAGAN. Besides, it even surpasses the performance of the pix2pix model. DAGAN gets excellent performance because the pix2pix model is built and optimized for this paired dataset with perfectly aligned image pairs. Besides, we could also observe from the region enclosed by black rectangles that DAGAN has a better performance in the conversion of roads, vegetations, and waterbodies over pix2pix. For the

performance of ADGAN and DAGAN, though they also used the attention mechanism in the model, it only captures the attention for foreground changes. The foreground and background attention mechanism helps in keeping good conversion performance both in foreground and background. DAGAN results have advantages over AGGAN in connections of roads, buildings, vegetation, and waterbodies, as shown in the figure above.

For the quantitative evaluation, we use the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) as the metric. PSNR is one of the most popular evaluation metrics for image conversion tasks which has reconstruction loss of images. It is defined as the equation below.

$$PSNR = 10 \times log_{10}\left(\frac{L^2}{\frac{1}{N}\sum_{i=1}^{N}\left(I_i - \hat{I}_i\right)^2}\right) \qquad (4-1)$$

where $I_i$ is the GT image and $\hat{I}_i$ is the result of conversion or reconstruction. $N$ is the number of pixels in them, and $L$ is the maximum pixel value. It is measured in dB via the $log_{10}$ function. For the quality, the higher the $PSNR$ is, the better quality of images is.

SSIM measures the structural similarity between images with independent comparison in terms of luminance, contrast, and structures from HSV space. It is defined as the equation below.

$$SSIM = \frac{2\left(\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \qquad (4-2)$$

where $\mu_x$and $\mu_y$ are local means for image $x$ and image $y$. $\sigma_x$ and $\sigma_y$ are the standard deviation of image $x$ and image $y$. $\sigma_{xy}$ is the cross-covariance between them. It is ranged in $(0,1)$. For the quality, the higher the $SSIM$ is, the better quality of images is.

PSNR is a metric that relies more on the MSE of pixel-level, which means that a high PSNR score would make sure that the generated images are similar with the GT in corresponding pixel values while the visual perception is not guaranteed. However, SSIM is similar to the evaluation system of human vision. A high SSIM score guarantees that the generated images and GT are visually similar in human eyes. These evaluation metrics have a complementary relationship to some extent, which is why we choose to use both of them in our evaluation part.

A quantitative comparison of generated images on Map2Aerial dataset between DAGAN, AGGAN, CycleGAN, and pix2pix models is shown in the table below. We calculated the PSNR and SSIM scores for all CycleGAN, pix2pix, AGGAN, and DAGAN test results. The results prove the qualitative evaluation results in **Figure 4-3** and show that DAGAN leads the first place both in pixel-level performance (PSNR) and visual generation performance (SSIM).

**Table 4-6**: Quantitative evaluation results for comparison of DAGAN on Map2Aerial dataset. Ours is the best one.

| Model Name | PSNR (dB) | SSIM |
|---|---|---|
| CycleGAN | 13.4 | 0.577 |
| Pix2pix | 14.2 | 0.597 |
| AGGAN | 14.4 | 0.604 |
| DAGAN (*ours*) | **14.9** | **0.619** |

#### 4.2.1.2 Ablation study with Map2Aerial Dataset

As DAGAN is a brand-new method proposed by our work. We also conducted an ablation study to figure out the influence of the main components and design and how they improve DAGAN performance.

We start with the ablation study on the dual-attention mechanism proposed by our work. The dual-attention mechanism consists of two independent attention modules that help in both foreground and background image conversion. We conducted both qualitative and quantitative studies on the Map2Aerial dataset and qualitative studies on the Color2IR dataset.

For the results on the Map2Aerial dataset, we compare the performance between a raw model without any attention mechanism (DAGAN-FA-BA), a model only with foreground attention but without background attention (DAGAN-BA), a model only with background attention but without foreground attention (DAGAN-FA), and DAGAN. During those tests, loss functions only needed to be modified in the raw model to match the network structure. The qualitative results are shown in the figure below.

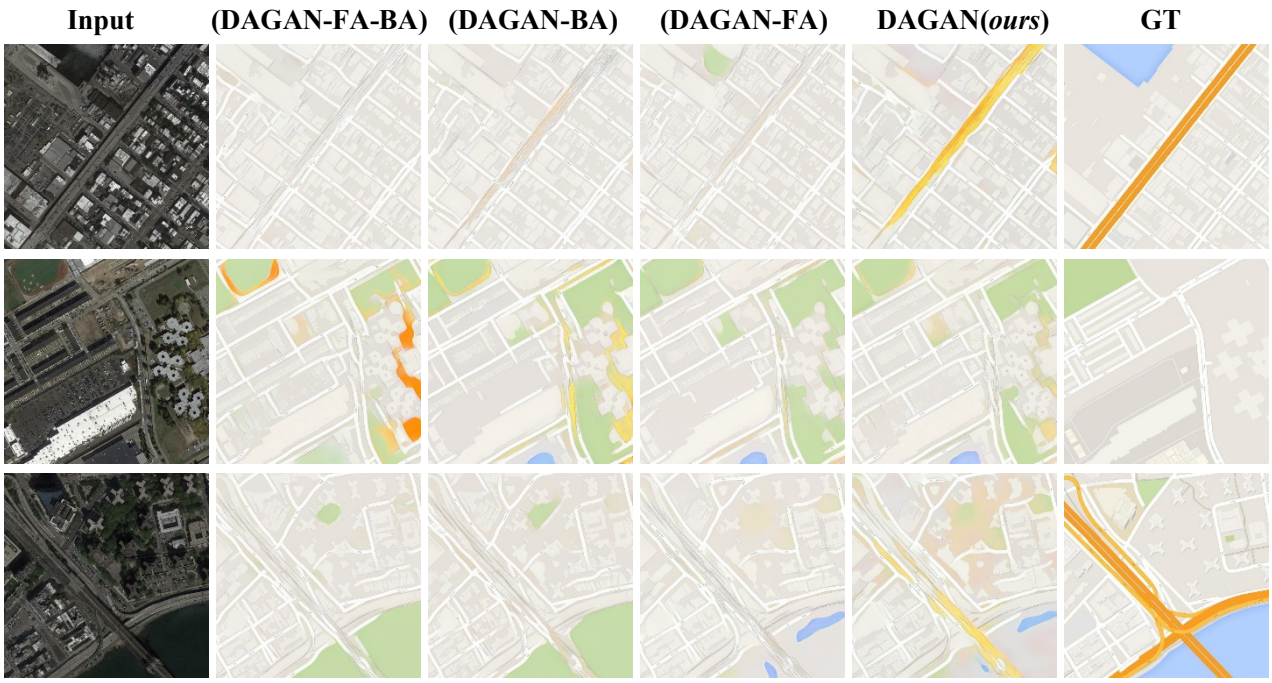


**Figure 4-4**: Images and conversion results for comparison in ablation study, the label above denotes the source of each column. Our DAGAN has the best performance among them.

As we can observe in **Figure 4-4**, the model without attention mechanism delivers the worst performance with loss in roads, vegetation and waterbodies areas, as it is not equipped with any attention module. There is no absolute winner between models with only one attention, as the FA model with foreground attention produced better results on the connection between roads and buildings. However, the BA model turns to have a good performance on the waterbodies and main roads, which are color-sensitive in this conversion. A possible reason for it is that the background attention module could capture those differences better. While DAGAN with dual-attention modules is the clear winner in this comparison, it captures the advantages in foreground attention and background attention, producing better results in main roads, waterbodies, and connections between different segments.

In general, the model without any attention mechanism somehow has the worst performance among them as it might produce some bad patterns for crowded buildings, roads, vegetation, or waterbodies. The results of the models with only one attention show that it allows them to generate images with fewer artifacts in vegetation areas. However, DAGAN has the dominant leading in all four models' performance in the accuracy of shapes in conversion and the class of conversions such as vegetation and waterbodies. As a result, it proves that the dual-attention mechanism proposed by us in DAGAN has a lot of positive effects on Map2Aerial conversion tasks.

For the quantitative ablation study, A comparison of generated images on the Map2Aerial dataset between DAGAN, a model with only one attention, and a model without attention is shown in the table below. We calculated the PSNR and SSIM scores for all test results. The results prove the qualitative evaluation result in **Figure 4-4** and show that DAGAN leads both in pixel-level performance (PSNR) and visual generation performance (SSIM).

**Table 4-7**: Quantitative evaluation results for comparison of DAGAN on Map2Aerial dataset. Ours is the best one in PSNR and SSIM metrics.

| Model Name | PSNR (dB) | SSIM |
|---|---|---|
| DAGAN-FA-BA | 13.7 | 0.585 |
| DAGAN-BA | 14.1 | 0.593 |
| DAGAN-FA | 14.4 | 0.599 |
| DAGAN (*ours*) | **14.9** | **0.619** |

In addition, the second part of the ablation study of DAGAN is about the new loss functions that we proposed. We use attention loss to solve the problem that attention masks could easily saturate to 1, which would ruin the generation of foreground and background content. The attention loss is shown in the equation below.

$$\mathcal{L}_{At}(A_x) = \sum_{w,h=1}^{W,H} |A_x(w+1,h,c) - A_x(w,h,c)| + |A_x(w,h+1,c) - A_x(w,h,c)| \qquad (4-3)$$

The ablation study aims to verify the influence of $\mathcal{L}_{At}$ in helping generate stable attention masks in DAGAN. We conducted both qualitative and quantitative studies on the Map2Aerial dataset.

For the results on the Map2Aerial dataset, we compare the performance between a RAW model without attention loss $\mathcal{L}_{At}$ (DAGAN-L) and DAGAN. During those tests, we only modified the total loss functions. The qualitative results are shown in the figure below.


**Figure 4-5**: Images and conversion results for comparison in ablation study, the label above denotes the source of each column. Our DAGAN has the best performance among them.

From **Figure 4-5**, we could observe that DAGAN outperformed the model without using our

proposed loss $\mathcal{L}_{At}$ in both different classes of segmentation and connection patterns, no matter in images mainly with buildings and roads or images with vegetations and waterbodies. As a result, we could see that $\mathcal{L}_{At}$ helps in the form of a stable transformation between the map and aerial domains by preventing saturation problems in the optimization process.

For the quantitative ablation study, A comparison of generated images on the Map2Aerial dataset between the DAGAN and DAGAN-L model is shown in the table below. We calculated the PSNR and SSIM scores for all test results from them. The results prove the qualitative evaluation result in **Figure 4-5** and show that DAGAN leads first place both in pixel-level performance (PSNR) and visual generation performance (SSIM).

**Table 4-8**: Quantitative ablation study for DAGAN on Map2Aerial dataset. The best one is in bold. Our DAGAN is the best one in PSNR and SSIM metrics.

| Model Name | PSNR (dB) | SSIM |
|---|---|---|
| DAGAN-L | 14.2 | 0.599 |
| DAGAN (*ours*) | **14.9** | **0.619** |

### 4.2.2 Video Semantic Segmentation Module

In the evaluation of the Video Semantic Segmentation Module, we are going to introduce the test results on the FS Segmentation dataset, together with a comparison with other state-of-the-art methods for image semantic segmentation tasks and video semantic segmentation tasks in accuracy and speed.

For the performance of the FS Segmentation dataset, we compare the results of the TD-Net that was used in our Video Semantic Segmentation module with two types of methods. The first type of model is used for image semantic segmentation, such as PSPNet [80] and DeepLab V3 [92]. The other type of models deals with video semantic segmentation tasks, like RGMP [107], SV-CNN [82], and SVS [84], and TD-Net [29], that we used in our module. **Figure 4-6** shows some of the visual samples generated by them. The labels at the left of the images show the source of one row of images, either Input, Ground Truth (GT), or generated by deep neural networks.

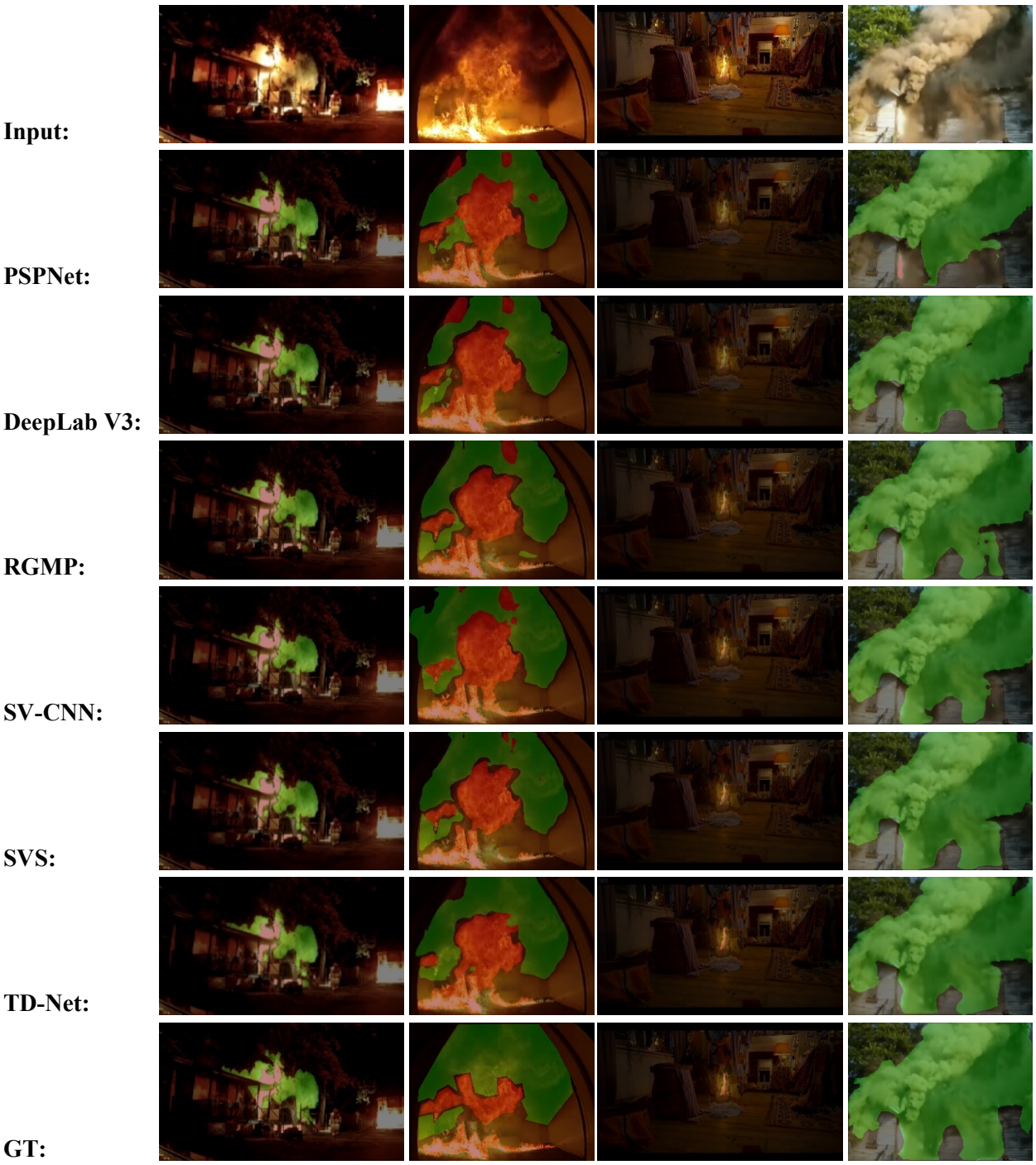


**Figure 4-6:** Samples of images for segmentation comparison, the label at left denotes the source of each row. Each column represents different scenes. There are 4 scenes in this figure. The red annotation is flame, and the green annotation is smoke. TD-Net is the one that used in our module.

From **Figure 4-6**, we could observe that in a dark environment like the images in the first column, both methods turned to give a good prediction of flame and smoke patterns. In contrast, the segmentation methods made for videos produced a better boundary for a smoke at the top. For images

in the second column, which is also from a video captured by firefighters but in a brighter condition, the performance of those models varies at this time. They all have an excellent general shape of smoke and flame pattern, while TD-Net and SVS give a more precise segmentation for the smoke details. For the images in the third column, video semantic segmentation methods show significant advantages over models based on image semantic segmentation methods. With the between frame information that is widely used in video semantic segmentation methods, they could easily capture the tiny flame patterns. For the images in the fourth column, every model gave similar segmentation results, which indicates that a large smoke pattern in a bright condition is somehow an easy pattern for segmentation.

Besides, we also compare the segmentation consistency between frames, which is also crucial for our tasks. **Figure 4-7** shows their results. The labels at the left of the images show the source of one row of images, either Input, Ground Truth (GT), or generated by deep neural networks. Each column of images is segmented on images captured at the same time. Among all the models, video semantic segmentation methods show dominant leading in the consistency between frames, and TD-Net is one of the best among them.

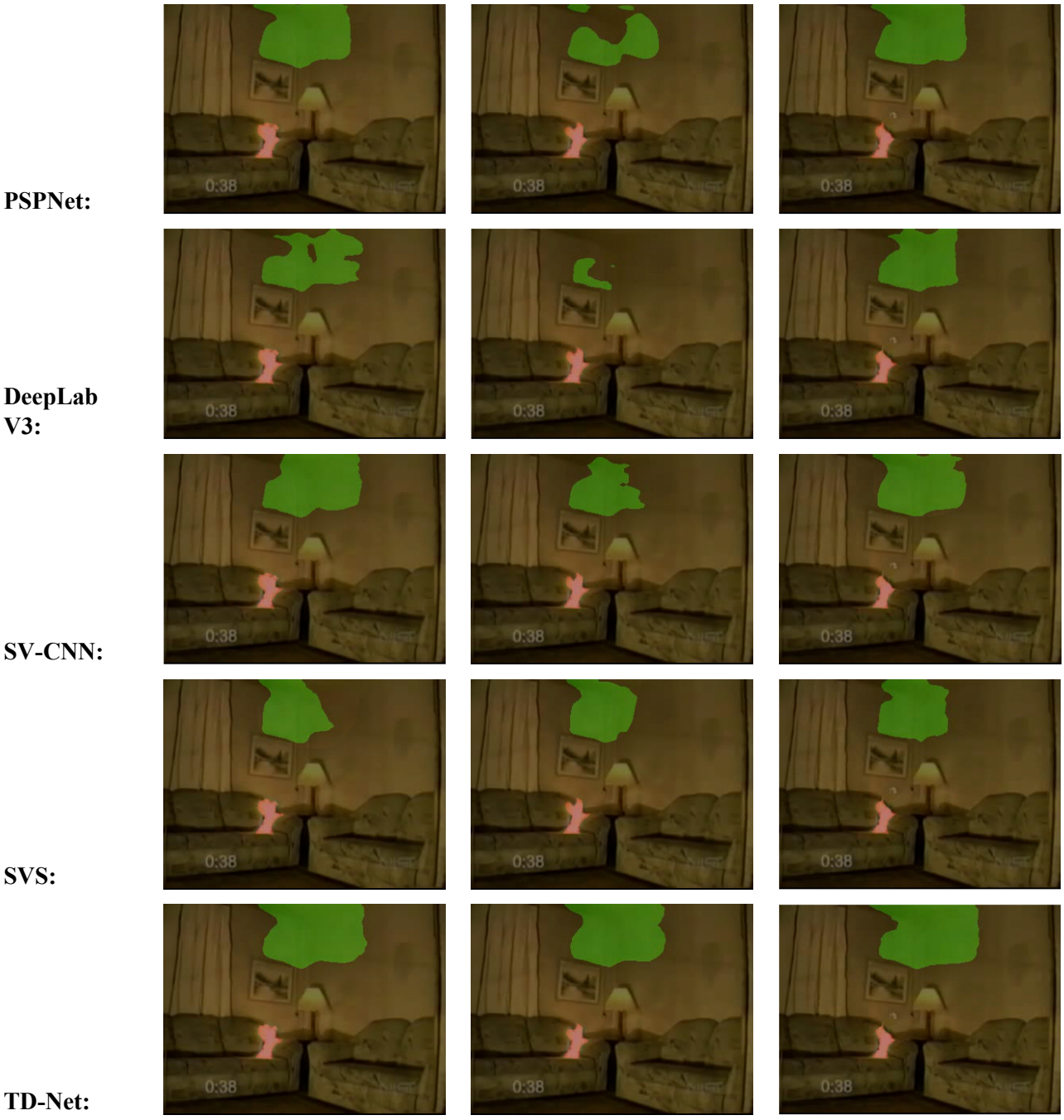


**Figure 4-7:** Images samples for consistency comparison, labels at left denotes the source of each row. Each column of images is from the same time. The sources of the three columns are three consecutive frames in the timeline. TD-Net is the one that used in our module.

Besides, we also conducted a quantitative study on those methods based on the Intersection of Union (IoU), mean Intersection of Union (mIoU), Accuracy (Acc), mean Accuracy (mAcc), and Speed scores.

IoU and mIoU is a popular metric used in the evaluation of segmentation tasks. They are both defined as the ratio of intersection and union of ground truth and predictions, while mIoU evaluates

on several classes. They could be formulated as the equation below.

$$IoU = \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}}$$
$$mIoU = \frac{1}{k+1} \sum_{i=1}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \quad (4-4)$$

where $p_{ij}$ denotes the samples that have a label of $i$ but segmented as $j$. $k+1$ is the total number of the class.

For Acc and mAcc, they are defined as the accuracy in pixel level, while mAcc evaluates on several classes. They are shown in the equation below.

$$Acc = \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}}$$
$$mAcc = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}} \quad (4-5)$$

where $p_{ij}$ denotes the samples that have a label of $i$ but are segmented as $j$. $k+1$ is the total number of the class.

Compared with IoU and mIoU, Acc and mAcc focused the accuracy on the pixel level, which could have advantages when different classes have a huge difference in the area. Besides, Acc and mAcc only calculated the impact of False Positive (FP) samples, where IoU and mIoU take both FP and False Negative (FN) into consideration.

Speed is a unique evaluation metric for video semantic segmentation. It aims to measure the ability of models to provide segmentation results for real-time. It is defined as the equation below.

$$Speed = \frac{T}{N} \quad (4-6)$$

where $N$ is the number of frames and $T$ is the processing time for it. It is measured in milliseconds per frame.

The comparison results based on mIoU, mAcc, and Speed are listed in the table below. We could find that TD-Net is in the leading place both in mIoU and mAcc matrices. Certain methods could surpass TD-Net in specific metrics like DeepLab V3 has a higher mIoU score than TD-Net. However, it has the best performance on average, both in accuracy and speed.

**Table 4-9**: Quantitative comparison for methods on FS Segmentation dataset. We measured three different metrics for them. TD-Net is the one that used in our module.

| Type of methods | Name of methods | mIoU (%) | mAcc (%) | Speed (ms/f) |
|---|---|---|---|---|
| Image Semantic Segmentation | PSPNet | 80.74 | 89.25 | 100 |
| | DeepLab V3 | **81.35** | **89.29** | 187 |
| Video Semantic Segmentation | RGMP | 80.69 | 88.43 | 77 |
| | SV-CNN | 80.8 | 89.13 | **59** |
| | SVS | 81.06 | 88.76 | 74 |
| | TD-Net | 81.29 | 89.28 | 70 |

Here is another table showing the extended information calculated by class. Though they are calculated for each class, the trend continues as TD-Net is also in leading places in different classes and metrics. In general, TD-Net has an excellent performance balance the accuracy and speed.

**Table 4-10**: Extended information of quantitative comparison for methods on FS Segmentation dataset. Metrics are measured in flame and smoke categories. TD-Net is the one that used in our module.

| Type of methods | Name of methods | IoU (%) | | Acc (%) | |
|---|---|---|---|---|---|
| | | Flame | Smoke | Flame | Smoke |
| Image Semantic Segmentation | PSPNet | 73.46 | 77.01 | 87.19 | 83.93 |
| | DeepLab V3 | 75.86 | **77.07** | 86.95 | **84.77** |
| Video Semantic Segmentation | RGMP | 76.78 | 74.95 | 86.3 | 83.06 |
| | SV-CNN | 76.13 | 75.56 | 87.66 | 83.87 |
| | SVS | 75.57 | 76.29 | 84.69 | 85.69 |
| | TD-Net | **76.85** | 75.97 | **87.89** | 83.91 |

### 4.2.3 Video Prediction Module

In the evaluation of the Video Prediction Module, we are going to introduce the test results on the FSP dataset. Besides, there is a comparison with other state-of-the-art methods for video prediction tasks in PSNR and SSIM scores.

**Figure 4-8** shows the qualitative comparison of video prediction performance among the FVP dataset. The labels at the left of the images show the source of one row of images. While, CNN-LP [90], Conv-LSTM [73], SVVP [100], AMC-GAN [104] are CNN-based, RNN-based, VAE-based, GAN-based video prediction methods, respectively. GT denotes the ground truth of frames. For each of the methods, they are given a set of frames as input, shown as the 'Initial Frames' in the first row. This figure includes seven predicted frames from each of the methods.

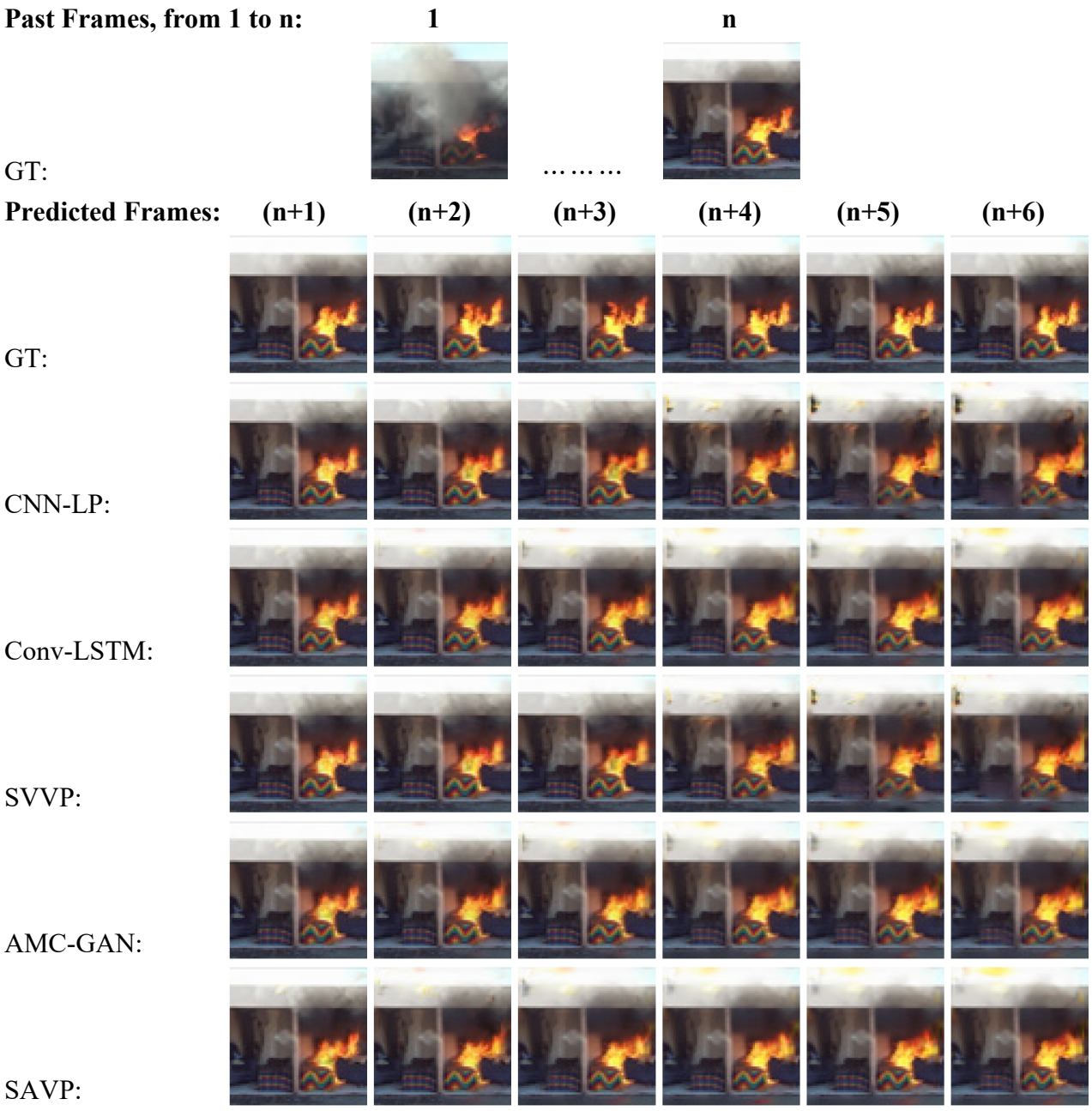


**Figure 4-8**: Examples of predicted images and past frames, the label at left denotes the source of each row. The number on top of images is the number of images. SAVP is the one that used in our module. It has the best performance.

From the figure above, we can observe that CNN-LP turns to capture the shape variation for the first few moments, and it is quickly getting blurred as time goes on. This phenomenon is especially obvious in the smoke area at the top right of the picture. The Conv-LSTM method has a similar situation as CNN-LP, while the shape of the flame and the smoke area is more apparent in the prediction in the last few frames. SVVP and AMC-GAN are generated-model-based approaches. However, SVVP produced frames with a better flame shape, which is more similar to the GT, while the blurry problem is more prominent, which is a typical issue in VAE-based video prediction

methods. AMC-GAN's performance is quite the opposite, with clear background and smoke area and a mixed area of flame in the last few predictions. SAVP is the best one among them, as it combines the benefits of VAE and GAN that could predict both clearer and plausible predictions even in the last few frames.

Besides, we also conducted a quantitative study on the models mentioned above, on PSNR and SSIM that varies with prediction time. The plots of them are shown in **Figure 4-9**. It shows the same trend as the qualitative study. SAVP has the best performance among all of them, whether in PSNR scores that measure the pixel difference or SSIM scores that are similar to human perceptual. AMC-GAN and SVVP show similar performance initially while failing to produce comparable results in the last few frames. As for CNN-LP and Conv-LSTM, they lost this race from the beginning of this competition.

The overall performance of SAVP proves its leading place in the video prediction task on the FVP dataset and the advantages of the combination of VAE and GAN.

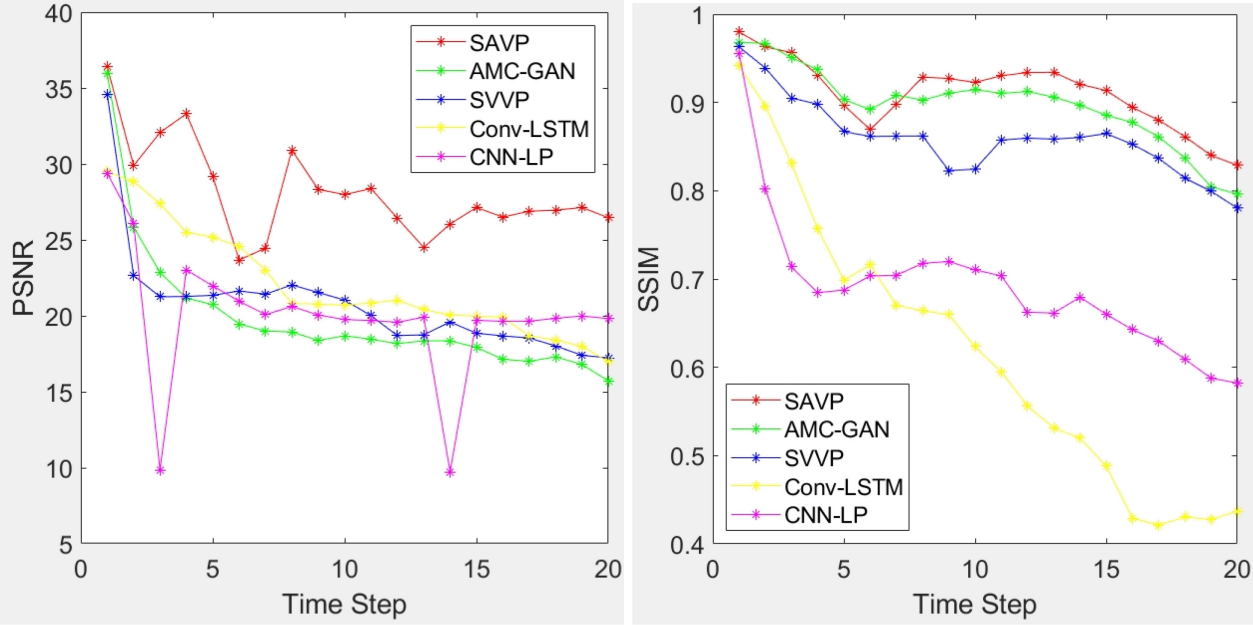


**Figure 4-9**: Plots of PSNR and SSIM scores with prediction time variation. Different lines represent the performance of different methods with the time developments. SAVP is the one that used in our module. It has the best performance.

As for the quantitative study on PSNR and SSIM scores somehow proves the visual results in the qualitative study as the scores are similar in the last few frames of prediction, whether in PSNR scores or SSIM scores. We could also observe the leadership of SAVP among them, while it is not a big lead this time compared with performance in the SAVP-V dataset.

In conclusion, we could find that the SAVP delivers the best performance among all other models in comparison. As a result, it proves our choice of it in the Video Prediction Module.

## 4.3 Evaluation of the entire system for flashover prediction

Flashover is a sudden fire propagation occurring in a room fire, which poses life-threatening hazards to firefighters. In this part, we test the performance of our system for flashover prediction. We first gave the dataset preparation and raw performance statistics. After that, we introduced a widely used metric in action prediction fields to evaluate flashover prediction. Finally, we compared the prediction performance of our system with other approaches in this field.

As some of the images for evaluation are confidential, more details of image results are available in NRC Report[1] [108].

The table below shows the raw statistics of our model.

**Table 4-11**: Raw statistics of flashover prediction performance of our system on the FP dataset, including prediction time, offsets, and the earliest forecast time.

| Source of video | Sequence name | Sequence length (s) | Flashover time (s) | Prediction time (s) | Offset (s) | Earliest Forecast time (s) |
|---|---|---|---|---|---|---|
| NRC | PRF-07 | 250 | 185 | 183 | -2 | 47 |
| | PRF-12 | 150 | 94 | 93 | -1 | 30 |
| | 08-SI-04 | 250 | 169 | 165 | -4 | 35 |
| | 14-SI-06 | 250 | 157 | 151 | -6 | 37 |
| | 21-SI-10 | 150 | 95 | 94 | -1 | 51 |
| | 23-SI-76 | 200 | 113 | 111 | -2 | 39 |
| | 31-SI-13 | 300 | 227 | 224 | -3 | 42 |
| NIST | NISTtest-1 | 50 | 22 | 15 | -7 | 10 |

In **Table 4-11**, we provide the prediction time of flashover by our proposed system (5$^{th}$ column), the offset between the flashover occurrence time and prediction (6$^{th}$ column), and the forecast time, which means the ability to tell the flashover happening in advance (7$^{th}$ column). We could observe a good performance both in prediction accuracy and forecast ability as they are all crucial to firefighters in compartment fires. Besides, the offset is all negative, and thus our system could always tell the flashover happening before it occurs. Considering that the offset is small in all cases, the flashover warning sent by our system should be effective and accurate. In comparison, the biggest offset comes from the evaluation of NISTtest-1. It is possible because we have not included any related visual or IR images in our system as well as each sub-module. In addition, the view angle and illumination conditions for the input video are also different.

A real-time analysis and prediction results could provide IR conversion, semantic segmentation, prediction of visual and IR frames, and statistical analysis of smoke and fire patterns fused by all sub-modules. Our judgment of flashover happening is based on the fusion of predicted future frames and

[1] 

temperature analysis, including shape of flame and smoke, Minimum/Average/Maximum temperature history data analysis, and flashover definition by NIST. For the final output, the system could provide a warning as well as *Estimated Time Arrival (ETA)* for the flashover in real-time.

Moreover, as there is no generic evaluation metric in the flashover prediction area, we also introduce an evaluation metric that is widely accepted in the action prediction field. Action prediction is similar to the prediction case of flashover as they both take video frames as input and give real-time predictions for the video. An example of an action prediction task is shown in the figure below. It shows a prediction task of human action with a partially observed video as input. It is the same case with real-time flashover prediction, as our system could only observe part of the video as time went on.

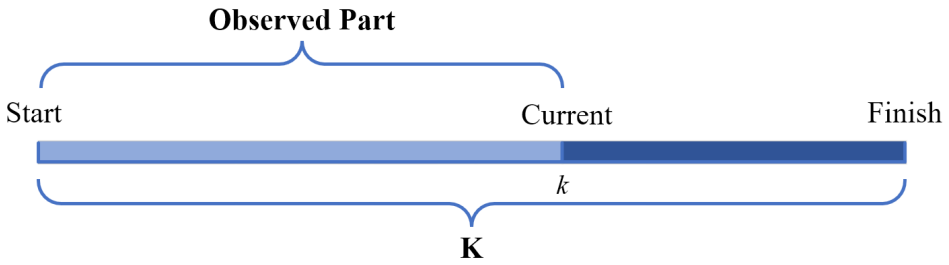


**Figure 4-10**: An example of a timeline of an action prediction task. $k$ is current time, $K$ is the total time.

A commonly used evaluation metric based on the observed part of the video in action prediction is the accuracy at observation ratio, as shown in **Figure 4-10**. The observation ratio is the ratio of the length of the observed part and the entire sequence, defined as the equation below.

$$r = \frac{k}{K} \tag{4-7}$$

where $k$ is the length of the observed part and $K$ is the length of the whole sequence, shown in **Figure 4-10**.

Using the accuracy at different observation ratios could eliminate the effect of different sequence lengths and thus provide a more objective evaluation.

When it comes to flashover prediction tasks, we modify the design of the observation ratio to fit the conditions in this task. Compared with the action that starts at the beginning of each sequence in the action dataset, flashovers start at the intermediate moment of the whole sequence. As the flashover prediction system aims to forecast the flashover before it happens, the period after flashover occurrence does not matter. As a result, the observation ratio in flashover prediction should be the ratio of the observed and un-flashover parts, formulated as the equation below.

$$r_f = \frac{t_c}{t_F} \tag{4-8}$$

where $r_f$ is the observation ratio in flashover prediction tasks, $t_c$ and $t_F$ is the current observation time and real time of flashover, shown in **Figure 4-11**.

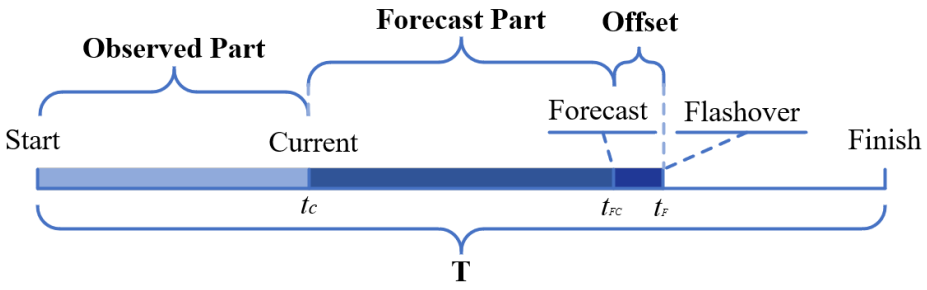


**Figure 4-11**: An illustration of time and period in a sequence of a flashover prediction. $r_f$ is the observation ratio in flashover prediction tasks, $t_c$ and $t_F$ is the current observation time and real time of flashover. $t_{FC}$ and $t_F$ is the predicted time of flashover and real time of flashover

Thus, we could measure the accuracy of flashover prediction at different observation ratio $r_f$.

Similar to the idea of accuracy at different observation ratio $r_f$, we also proposed a new forecast Accuracy ($FA$) score at different observation ratios $r$ for the evaluation of flashover forecasting. It could be formulated as equation 7-9.

$$FA = 1 - \frac{|t_F - t_{FC}|_{abs}}{t_F} \qquad (4-9)$$

where $t_{FC}$ and $t_F$ is the predicted time of flashover and real time of flashover, shown in **Figure 4-11**.

$FA$ with observation ratio $r_f$ would help eliminate the impact of the sequence length. As a result, we compare the performance of our system with other state-of-the-art flashover prediction systems proposed in recent years based on the two metrics introduced above. The results are shown in **Table 4-12**.

**Table 4-12**: Comparison of flashover prediction performance with other models. Our system has the best performance among different metrics.

| Name of model | Acc@$r_f$ | | FA@$r_f$ | |
|---|---|---|---|---|
| | **Acc@0.5** | **Acc@1** | **FA@0.5** | **FA@1** |
| *Dexters et al.* [109] | - | 0.91 | - | - |
| *Fliszkiewicz et al.* [110] | - | 0.6569 | - | - |
| *Yap et al.* [55] | - | 0.94 | - | - |
| *Lee et al.* [111] | - | 0.92 | - | - |
| *Fu et al.* [54] | 0.761 | **1** | 0.681 | 0.813 |
| *Yun et al.* [56] | - | **1** | - | 0.92 |
| *ours* | **0.875** | **1** | **0.813** | **0.945** |

Though some of the models listed in the table do not use the same dataset for the test, all the datasets are at least from a room environment similar to a room shown in the figure below.

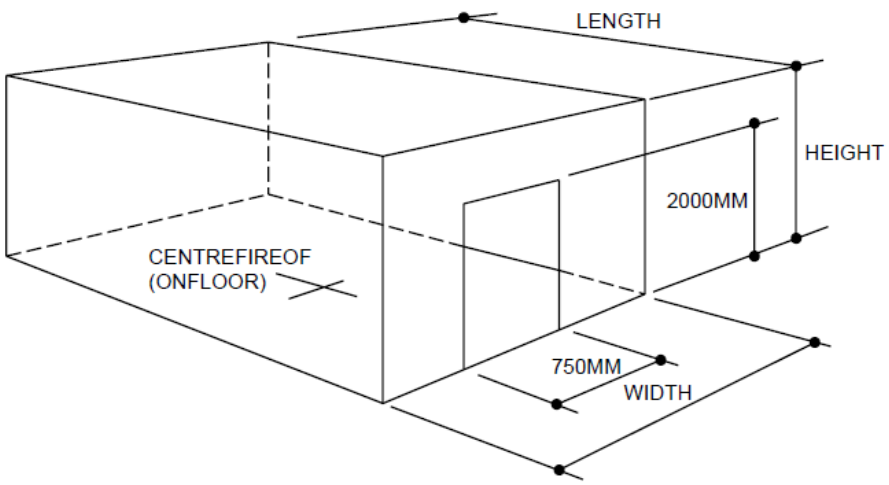


**Figure 4-12**: Fire compartment setting for flashover experiments, from [111].

We could observe from the table that most of the models only provide accuracy at $r_f = 1$, which means those models are only a 'classification' task given the entire sequence. Besides, some of them provided the forecast time and prediction time so that they are capable of doing the forecast job, and we could evaluate the accuracy at $r_f = 0.5$ and $FA$ at $r_f = 1$ or $r_f = 0.5$. However, the prediction performance of our system is in the top places among all other models in accuracy and $FA$ at $r_f = 0.5$ or $r_f = 1$. While some methods might show comparable results at one particular metric, our model shows a much powerful forecast ability for flashover, as shown in metrics **Acc@0.5 and FA@0.5**. It points out that our system has a high flashover prediction accuracy and high flashover forecast ability as well.

Among all the systems in the table, the one proposed by Yun et al. [56] is the most related work. The difference between their system and ours is that we extract information from RGB images to generate semantic area information, such as smoke and flame. That area information is vital because they are crucial parts of a fire that could show fire status and development. For example, one of the criteria used by both of us for flashover occurrence is the smoke area reaching 600 °C or above. With the help of semantic information from RGB images, our system could precisely collect the smoke area's temperature information. However, their methods would calculate the temperature based on the whole region, which would reduce the accuracy.

The results also prove it. Although we got the same performance in **Acc@1**, our methods surpass their methods in metric **FA@1** with a lead of 2.5%. It demonstrates that the semantic information extracted from RGB images could help achieve higher accuracy in flashover prediction.

# Chapter 5. Conclusion

## 5.1 Conclusion

Flame and smoke analysis are some of the bases for modern fire science. The critical part of analyzing them is thermal information, which could be captured only by IR cameras. As for the existing research in this field, they mainly focused on simple mathematic models like regression with fire knowledge. However, those methods have limitations in firefighting usage, as IR cameras are expensive. Deep learning and deep neural networks have been proved to be effective in extracting information from images and videos, which reduces the hardware requirement and enables a system based on RGB cameras. The combination of RGB images and deep learning methods for flame and smoke analysis systems can not only be portable for firefighters but also have high accuracy.

(a) About the Flame and Smoke Analysis System

In this thesis, we propose a new system that is able to analyze flame and smoke only based on RGB images. It has a novel structure of 4 sub-modules for different tasks. It is also a hybrid one with fire safety knowledge and deep learning techniques. These sub-modules can be combined and used for specific goals such as smoke detection, flame detection, or sudden fire propagation.

The Color2IR module is one of the most important modules in the system. The key part of it is DAGAN proposed by us. In DAGAN, we apply an attention module capable of analyzing both foreground and background attention, which helps form a sharp foreground and a clear background of images.

Next, Video Semantic Segmentation Module helps in extracting flame and smoke areas from the scene in the RGB video frames. It is based on a well-known deep neural network: TD-Net, which balances segmentation accuracy and speed. Besides, we innovated to use synthetic RGB video data generated and captured from 3D modeling software for data augmentation to improve accuracy.

After that, a Video Prediction Module takes the RGB video frames and IR frames as input and produces predictions of the subsequent frames of their scenes. The key part of it is an existing deep neural network called SAVP, which takes advantage of the visually plausible results from GAN and diverse output from VAE.

Finally, a Fire Knowledge Analysis Module predicts if a flashover is coming or not. It is based on fire knowledge criteria such as thermal information extracted from IR images, temperature increase rate, flashover occurrence temperature, and increase rate of lowest temperature. That information comes from current RGB video and IR frames, fire and smoke segmentation, and next predicted frames of RGB video and IR frame.

For the contributions and innovations in our work, we propose a novel network, DAGAN, with foreground and background attention in the image conversion. It helps in reducing the hardware device requirement and achieving high accuracy for flashover prediction. Besides, we also combine thermal information from IR images and segmentation information from RGB images in our system

in order to analyze flame and smoke. This combination will improve the accuracy in analysis. We also design a hybrid structure for our entire system, which combines several deep neural networks and a knowledge-based system to achieve high accuracy. Moreover, data augmentation is used on the Video Semantic Segmentation Module by introducing synthetic video data in the training process.

(b) About experiments and evaluation

Subsequently, we analyze the performance of our whole model as well as each sub-module on several datasets.

Firstly, we evaluate the performance of DAGAN on an ordinary dataset for paired image conversion tasks. It shows that it is superior to other existing methods such as pix2pix, CycleGAN, or AGGAN quantitatively and qualitatively. The evaluation results on Color2IR conversion tasks show that DAGAN has a leading place. For the ablation study, we evaluate the attention mechanism as well as the attention loss design, and each part shows its advantages. Then, we also compare the performance of other modules on their tasks with other prevailing models. Their top performance further proves our choice.

Finally, we evaluate our system as a whole part for flashover prediction. We show that our proposed system gives a promising performance on flashover prediction tasks. In addition, we also introduce a set of new metrics inspired by the action prediction evaluation so that we can compare the results of our model with other existing flashover prediction models. The overall comparison shows that our system delivers higher accuracy and is capable of giving earlier forecasts for flashover occurrence at the same time.

## 5.2 Future work

There is still room for further improvement in our system. One of the possible works is to explore the applicability of each module's flame and smoke hazard identification. For example, the prediction quality of the sub-module Color2IR is relatively low because the contextual information in IR images is different from that in color images. Further research is necessary to improve the image predictions (e.g., smoke/flame development). Therefore, the way that our module extracts and uses the temperature information could be further explored.

As demonstrated, the sub-modules can be combined and built for the detection or analysis of fire development. However, the testing is complex because most of the existing research in this field would like to set up their experiments and collect data. Therefore, for a thorough comparison between different models, A benchmark dataset would boost research progress in this area.

# References


[1] Karter, M. (1998). Fire Loss in the United States During.

[2] Bowyer, M. E., Miles, V., Baldwin, T. N., & Hales, T. R. (2016). Preventing deaths and injuries of fire fighters during training exercises.

[3] Butler, C., Marsh, S., Domitrovich, J. W., Helmkamp, J. J. J. o. o., & hygiene, e. (2017). Wildland firefighter deaths in the United States: A comparison of existing surveillance systems. *14*(4), 258-270.

[4] Opert, K. M., Lock, A. J., Bundy, M. F., Johnsson, E. L., Hwang, C., Hamins, A. P., . . . Lee, K.-Y. (2012). Experimental Study of the Three Dimensional Internal Structure of Underventilated Compartment Fires in an ISO 9705 Room.

[5] C, W. (2020). How smoke detectors work. *ExplainThatStuff*.

[6] Geetha, S., Abhishek, C., & Akshayanat, C. J. F. T. (2021). Machine Vision Based Fire Detection Techniques: A Survey. *57*(2), 591-623.

[7] Russo, A. U., Deb, K., Tista, S. C., & Islam, A. (2018). *Smoke detection method based on LBP and SVM from surveillance camera.* Paper presented at the 2018 International conference on computer, communication, chemical, material and electronic engineering (IC4ME2).

[8] Tang, T., Dai, L., & Yin, Z. (2017). *Smoke image recognition based on local binary pattern.* Paper presented at the 2017 5th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2017).

[9] Yuan, F. J. F. s. j. (2011). Video-based smoke detection with histogram sequence of LBP and LBPV pyramids. *46*(3), 132-139.

[10] Liu, Z., Yang, X., Liu, Y., & Qian, Z. J. I. A. (2019). Smoke-detection framework for high-definition video using fused spatial-and frequency-domain features. *7*, 89687-89701.

[11] Gao, Y., & Cheng, P. J. F. t. (2019). Forest fire smoke detection based on visual smoke root and diffusion model. *55*(5), 1801-1826.

[12] Jian, W., Wu, K., Yu, Z., & Chen, L. (2018). *Smoke regions extraction based on two steps segmentation and motion detection in early fire.* Paper presented at the MIPPR 2017: Pattern Recognition and Computer Vision.

[13] Jinlan, L., Lin, W., Ruliang, Z., Chengquan, H., & Yan, R. (2016). *A method of fire and smoke detection based on surendra background and gray bitmap plane algorithm.* Paper presented at the 2016 8th international conference on information technology in medicine and education (ITME).

[14] Jia, Y., Yuan, J., Wang, J., Fang, J., Zhang, Q., & Zhang, Y. J. F. t. (2016). A saliency-based method for early smoke detection in video sequences. *52*(5), 1271-1292.

[15] Luo, S., Yan, C., Wu, K., & Zheng, J. J. F. S. J. (2015). Smoke detection based on condensed image. *75*, 23-35.

[16] Razmi, S. M., Saad, N., & Asirvadam, V. S. (2010). *Vision-based flame analysis using motion and edge detection.* Paper presented at the 2010 international conference on intelligent and advanced systems.

[17] Wu, X., Lu, X., & Leung, H. J. S. (2018). A video based fire smoke detection using robust AdaBoost. *18*(11), 3780.

[18] Lee, Y., Kim, T., & Shim, J. J. J. o. M. I. S. (2017). Smoke detection system research using fully connected method based on adaboost. *4*(2), 79-82.

[19] Zhang, F., Qin, W., Liu, Y., Xiao, Z., Liu, J., Wang, Q., . . . Applications. (2020). A Dual-Channel convolution neural network for image smoke detection. 1-17.

[20] Gu, K., Xia, Z., Qiao, J., & Lin, W. J. I. T. o. M. (2019). Deep dual-channel neural network for image-based smoke detection. *22*(2), 311-323.

[21] Di Lascio, R., Greco, A., Saggese, A., & Vento, M. (2014). *Improving fire detection reliability by a combination of videoanalytics.* Paper presented at the International Conference Image Analysis and Recognition.

[22] Pundir, A. S., & Raman, B. J. F. t. (2019). Dual deep learning model for image based smoke detection. *55*(6), 2419-2442.

[23] Yin, M., Lang, C., Li, Z., Feng, S., Wang, T. J. M. T., & Applications. (2019). Recurrent convolutional network for video-based smoke detection. *78*(1), 237-256.

[24] Filonenko, A., Kurnianggoro, L., & Jo, K.-H. (2017). *Smoke detection on video sequences using convolutional and recurrent neural networks.* Paper presented at the International Conference on Computational Collective Intelligence.

[25] Kim, T., Cha, M., Kim, H., Lee, J. K., & Kim, J. (2017). *Learning to discover cross-domain relations with generative adversarial networks.* Paper presented at the International Conference on Machine Learning.

[26] Yi, Z., Zhang, H., Tan, P., & Gong, M. (2017). *Dualgan: Unsupervised dual learning for image-to-image translation.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. J. a. p. a. (2017). Attention is all you need.

[28] Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). *Unpaired image-to-image translation using cycle-consistent adversarial networks.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

[29] Hu, P., Caba, F., Wang, O., Lin, Z., Sclaroff, S., & Perazzi, F. (2020). *Temporally distributed networks for fast video semantic segmentation.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[30] Hinton, G., Vinyals, O., & Dean, J. J. a. p. a. (2015). Distilling the knowledge in a neural network.

[31] Lee, A. X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., & Levine, S. J. a. p. a. (2018). Stochastic adversarial video prediction.

[32] Babrauskas, V. J. F. T. (1980). Estimating room flashover potential. *16*(2), 94-103.

[33] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. J. P. o. t. I. (1998). Gradient-based learning applied to document recognition. *86*(11), 2278-2324.

[34] Krizhevsky, A., Sutskever, I., & Hinton, G. E. J. A. i. n. i. p. s. (2012). Imagenet classification with deep convolutional neural networks. *25*, 1097-1105.

[35] He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[36] He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Identity mappings in deep residual networks.* Paper presented at the European conference on computer vision.

[37] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). *Going deeper with convolutions.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[38] Simonyan, K., & Zisserman, A. J. a. p. a. (2014). Very deep convolutional networks for large-scale image recognition.

[39] Karlik, B., Olgac, A. V. J. I. J. o. A. I., & Systems, E. (2011). Performance analysis of various activation functions in generalized MLP architectures of neural networks. *1*(4), 111-122.

[40] Ramachandran, P., Zoph, B., & Le, Q. V. J. a. p. a. (2017). Searching for activation functions.

[41] Sibi, P., Jones, S. A., Siddarth, P. J. J. o. t., & technology, a. i. (2013). Analysis of different activation functions using back propagation neural networks. *47*(3), 1264-1268.

[42] Zhang, Z., & Sabuncu, M. R. J. a. p. a. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels.

[43] Taqi, A. M., Awad, A., Al-Azzo, F., & Milanova, M. (2018). *The impact of multi-optimizers and data augmentation on TensorFlow convolutional neural network performance.* Paper presented at the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR).

[44] Bello, I., Zoph, B., Vasudevan, V., & Le, Q. V. (2017). *Neural optimizer search with reinforcement learning.* Paper presented at the International Conference on Machine Learning.

[45] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010* (pp. 177-186): Springer.

[46] Hinton, G., Srivastava, N., & Swersky, K. J. C. o. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *14*(8).

[47] Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). *On the importance of initialization and momentum in deep learning.* Paper presented at the International conference on machine learning.

[48] Duchi, J., Hazan, E., & Singer, Y. J. J. o. m. l. r. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *12*(7).

[49] Kingma, D. P., & Ba, J. J. a. p. a. (2014). Adam: A method for stochastic optimization.

[50] Xu, Z., Wanguo, W., Xinrui, L., Bin, L., & Yuan, T. (2019). *Flame and smoke detection in substation based on wavelet analysis and convolution neural network.* Paper presented at the Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence.

[51] Zhong, Z., Wang, M., Shi, Y., Gao, W. J. S., Image, & Processing, V. (2018). A convolutional neural network-based flame detection method in video sequence. *12*(8), 1619-1627.

[52] Kim, B., & Lee, J. J. A. S. (2019). A video-based fire detection using deep learning models. *9*(14), 2862.

[53] Hu, Y., Lu, X. J. M. T., & Applications. (2018). Real-time video fire smoke detection by utilizing spatial-temporal ConvNet features. *77*(22), 29283-29301.

[54] Fu, E. Y., Tam, W. C., Wang, J., Peacock, R., Reneke, P., Ngai, G., . . . Cleary, T. (2021). Predicting Flashover Occurrence using Surrogate Temperature Data.

[55] Yap, K. S., Lee, C. P. L. E. W., & Saleh, J. M. (2009). *Development and Application of An Enhanced ART-Based Neural Network.* Paper presented at the The International Conference on Man-Machine Systems.

[56] Yun, K., Bustos, J., & Lu, T. J. E. I. (2018). Predicting rapid fire growth (flashover) using conditional generative adversarial networks. *2018*(9), 127-121-127-124.

[57] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Bing, X., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). *Generative adversarial nets.* Paper presented at the International Conference on Neural Information Processing Systems.

[58] Radford, A., Metz, L., & Chintala, S. J. a. p. a. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks.

[59] Yeh, R. A., Chen, C., Yian Lim, T., Schwing, A. G., Hasegawa-Johnson, M., & Do, M. N. (2017). *Semantic image inpainting with deep generative models.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[60] Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). *Image-to-image translation with conditional adversarial networks.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[61] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., . . . Wang, Z. (2017). *Photo-realistic single image super-resolution using a generative adversarial network.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[62] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., . . . Change Loy, C. (2018). *Esrgan: Enhanced super-resolution generative adversarial networks.* Paper presented at the Proceedings of the European Conference on Computer Vision (ECCV) Workshops.

[63] Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein generative adversarial networks.* Paper presented at the International conference on machine learning.

[64] Mirza, M., & Osindero, S. J. a. p. a. (2014). Conditional generative adversarial nets.

[65] Tang, H., Xu, D., Sebe, N., & Yan, Y. (2019). *Attention-guided generative adversarial networks for unsupervised image-to-image translation.* Paper presented at the 2019 International Joint Conference on Neural Networks (IJCNN).

[66] Bourlard, H., & Kamp, Y. J. B. c. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *59*(4), 291-294.

[67] Kingma, D. P., & Welling, M. J. a. p. a. (2013). Auto-encoding variational bayes.

[68] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. J. a. p. a. (2015). Adversarial autoencoders.

[69] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). *Extracting and composing robust features with denoising autoencoders.* Paper presented at the Proceedings of the 25th international conference on Machine learning.

[70] Wold, S., Esbensen, K., Geladi, P. J. C., & systems, i. l. (1987). Principal component analysis. *2*(1-3), 37-52.

[71] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. J. a. p. a. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

[72] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

[73] Cheng, J., Dong, L., & Lapata, M. J. a. p. a. (2016). Long short-term memory-networks for machine reading.

[74] Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). *Self-attention generative adversarial networks.* Paper presented at the International conference on machine learning.

[75] Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., Terzopoulos, D. J. I. T. o. P. A., & Intelligence, M. (2021). Image segmentation using deep learning: A survey.

[76] Otsu, N. J. I. t. o. s., man,, & cybernetics. (1979). A threshold selection method from gray-level histograms. *9*(1), 62-66.

[77] Nock, R., Nielsen, F. J. I. T. o. p. a., & intelligence, m. (2004). Statistical region merging. *26*(11), 1452-1458.

[78] Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully convolutional networks for semantic segmentation.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[79] Noh, H., Hong, S., & Han, B. (2015). *Learning deconvolution network for semantic segmentation.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

[80] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). *Pyramid scene parsing network.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[81] Chen, L.-C., Yang, Y., Wang, J., Xu, W., & Yuille, A. L. (2016). *Attention to scale: Scale-aware semantic image segmentation.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[82] Gadde, R., Jampani, V., & Gehler, P. V. (2017). *Semantic video cnns through representation warping.* Paper presented at the Proceedings of the IEEE International Conference on Computer Vision.

[83] Yu, C., Ma, X., Ren, J., Zhao, H., & Yi, S. (2020). *Spatio-Temporal Graph Transformer Networks for Pedestrian Trajectory Prediction.* Paper presented at the European Conference on Computer Vision.

[84] Nilsson, D., & Sminchisescu, C. (2018). *Semantic video segmentation by gated recurrent flow propagation.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[85] Li, Y., Shi, J., & Lin, D. (2018). *Low-latency video semantic segmentation.* Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[86] Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J. A., Orts-Escolano, S., Garcia-Rodriguez, J., . . . Intelligence, M. (2020). A review on deep learning techniques for

video prediction.

[87] Lecun, Y., Bengio, Y., & Hinton, G. J. N. (2015). Deep learning. *521*(7553), 436.

[88] Chen, X., Wang, W., Wang, J., & Li, W. (2017). *Learning object-centric transformation for video prediction.* Paper presented at the Proceedings of the 25th ACM international conference on Multimedia.

[89] Mathieu, M., Couprie, C., & LeCun, Y. J. a. p. a. (2015). Deep multi-scale video prediction beyond mean square error.

[90] Denton, E., Chintala, S., Szlam, A., & Fergus, R. J. a. p. a. (2015). Deep generative image models using a laplacian pyramid of adversarial networks.

[91] Yu, F., Koltun, V., & Funkhouser, T. (2017). *Dilated residual networks.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[92] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L. J. I. t. o. p. a., & intelligence, m. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *40*(4), 834-848.

[93] Luo, W., Li, Y., Urtasun, R., & Zemel, R. J. a. p. a. (2017). Understanding the effective receptive field in deep convolutional neural networks.

[94] Terwilliger, A., Brazil, G., & Liu, X. (2019). *Recurrent flow-guided semantic forecasting.* Paper presented at the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV).

[95] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. J. a. p. a. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting.

[96] Lotter, W., Kreiman, G., & Cox, D. J. a. p. a. (2015). Unsupervised learning of visual structure using predictive generative networks.

[97] Hochreiter, S., & Schmidhuber, J. J. N. c. (1997). Long short-term memory. *9*(8), 1735-1780.

[98] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. J. a. p. a. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation.

[99] Bhattacharyya, A., Fritz, M., & Schiele, B. J. a. p. a. (2018). Bayesian prediction of future street scenes using synthetic likelihoods.

[100] Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., & Levine, S. J. a. p. a. (2017). Stochastic variational video prediction.

[101] Liang, X., Lee, L., Dai, W., & Xing, E. P. (2017). *Dual motion gan for future-flow embedded video prediction.* Paper presented at the proceedings of the IEEE international conference on computer vision.

[102] Denton, E., & Fergus, R. (2018). *Stochastic video generation with a learned prior.* Paper presented at the International Conference on Machine Learning.

[103] Hu, Z., & Wang, J. (2019). *A novel adversarial inference framework for video prediction with action control.* Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.

[104] Jang, Y., Kim, G., & Song, Y. (2018). *Video prediction with appearance and motion*

*conditions.* Paper presented at the International Conference on Machine Learning.

[105] Bwalya, A. G., Eric; Lougheed, Gary; Kashef, Ahmed. (2014). Characterization of fires in multi-suite residential dwellings: final project report: part 1-A compilation of post-flashover room fire test data. *Research Report (National Research Council of Canada. Construction)*. doi:https://doi.org/10.4224/21275340

[106] NIST, M. H., M. Bundy. (2017). Dry Tree Fire Hazard.

[107] Oh, S. W., Lee, J.-Y., Sunkavalli, K., & Kim, S. J. (2018). *Fast video object segmentation by reference-guided mask propagation.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[108] Li, Y. a. K., Yoon. (2021). *Development of a Hybrid Algorithm to predict room fire flashovers based on Vision Data*. Retrieved from

[109] Dexters, A., Leisted, R. R., Van Coile, R., Welch, S., Jomaas, G. J. F., & Materials. (2020). Testing for knowledge: Application of machine learning techniques for prediction of flashover in a 1/5 scale ISO 13784-1 enclosure.

[110] Fliszkiewicz, M., Krasuski, A., & Krenski, K. (2014). *Evaluation of a Heat Release Rate based on Massively Generated Simulations and Machine Learning Approach.* Paper presented at the FedCSIS (Position Papers).

[111] Lee, E. W., Lee, Y., Lim, C., & Tang, C. Y. J. A. e. i. (2006). Application of a noisy data classification technique to determine the occurrence of flashover in compartment fires. *20*(2), 213-222.