6-6-2019

Multimodal Data Analytics and Fusion for Data Science

Haiman Tian
Florida International University, htian005@fiu.edu

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

MULTIMODAL DATA ANALYTICS AND FUSION FOR DATA SCIENCE

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Haiman Tian

2019

To: Dean John L. Volakis
College of Engineering and Computing

This dissertation, written by Haiman Tian, and entitled Multimodal Data Analytics and Fusion for Data Science, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Xudong He

Sitharama S. Iyengar

Jainendra K. Navlakha

Keqi Zhang

Shu-Ching Chen, Major Professor

Date of Defense: June 6, 2019

The dissertation of Haiman Tian is approved.

Dean John L. Volakis
College of Engineering and Computing

Andrés G. Gil
Vice President for Research and Economic Development
and Dean of the University Graduate School

Florida International University, 2019

DEDICATION

To my parents and the memory of my grandfather.

ACKNOWLEDGMENTS

First of all, I would like to dedicate my utmost gratitude to my advisor Professor Shu-Ching Chen for his invaluable guidance, encouragement, patience, and support through so many years of research. In addition, I would also like to thank Professor Sitharama S. Iyengar, Jainendra K. Navlakha, Xudong He of the School of Computing and Information Sciences, and Professor Keqi Zhang of the Department of Environmental Studies and International Hurricane Research Center for the suggestions they provided.

Secondly, my thanks would go to the friends and colleagues from the Distributed Multimedia Information Systems (DMIS) Laboratory at FIU and the Data Mining, Database & Multimedia (DDM) Research Group at University of Miami, in particular, Hsin-Yu Ha, Yimin Yang, Samira Pouyanfar, María Presa Reyes, Hector Cen Zheng, and Yudong Tao. Special thanks also goes to all the people who I have met and worked with during these years.

Last but not least, I am extremely grateful for the unconditional support and love from my parents. They were always encouraging me with their best wishes. I would never have been able to finish my dissertation without their support and encouragement.

This research work is partially supported by NSF CNS-1461926, DHS's VACCINE Center under Award Number 2009-ST- 061-CI0001, Florida Public Hurricane Loss Model, and FIU's Dissertation Year Fellowship (DYF).

ABSTRACT OF THE DISSERTATION

MULTIMODAL DATA ANALYTICS AND FUSION FOR DATA SCIENCE

by

Haiman Tian

Florida International University, 2019

Miami, Florida

Professor Shu-Ching Chen, Major Professor

Advances in technologies have rapidly accumulated a zettabyte ($\approx 10^{21}$) of "new" data every two years. The huge amount of data have a powerful impact on various areas in science and engineering and generates enormous research opportunities, which calls for the design and development of advanced approaches in data analytics. Given such demands, data science has become an emerging hot topic in both industry and academia, ranging from basic business solutions, technological innovations, and multidisciplinary research to political decisions, urban planning, and policymaking. Within the scope of this dissertation, a multimodal data analytics and fusion framework is proposed for data-driven knowledge discovery and cross-modality semantic concept detection. The proposed framework can explore useful knowledge hidden in different formats of data and incorporate representation learning from data in multimodalities, especially for disaster information management. First, a Feature Affinity-based Multiple Correspondence Analysis (FA-MCA) method is presented to analyze the correlations between low-level features from different features, and an MCA-based Neural Network (MCA-NN) is proposed to capture the high-level features from individual FA-MCA models and seamlessly integrate the semantic data representations for video concept detection. Next, a genetic algorithm-based approach is presented for deep neural network selection. Furthermore, the improved genetic algorithm is integrated with deep neural networks to generate populations for producing optimal deep representation learning models. Then, the multimodal

deep representation learning framework is proposed to incorporate the semantic representations from data in multiple modalities efficiently. At last, fusion strategies are applied to accommodate multiple modalities. In this framework, cross-modal mapping strategies are also proposed to organize the features in a better structure to improve the overall performance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background and Introduction

During the Internet era, the volume of real-time digital data generated world wide has grown exponentially. About 70% of it includes multimedia content carrying a variety of valuable visual, aural, and textual information. These digital data streams can be accessed by a host of devices, including mobile phones and tablets, sensor-equipped infrastructure, vehicles, and intelligent household appliances. The availability and accessibility of this vast amount of "new" data has prompted a big data revolution. Using those data wisely can have an enormous impact on research, technological innovation, and even policymaking. Data Science (DS), a multidisciplinary branch of science seeking to leverage big data effectively, has exploded in popularity among academic researchers and industry experts since 2000. DS is not only changing the world but also improving our everyday lives. It offers groundbreaking ways to understand the most difficult problems and devise urgently needed solutions within fields as disparate as economics, urban planning, public health, and political science. DS has played an especially important role in crisis response and recovery in recent disaster scenarios with promising results. During a catastrophe, informed decision-making is crucial to prevent further damage and ultimately to save lives. Based on simulation results, DS can minimize the damage caused by a disaster by utilizing big data as a convincing early warning tool. To advance these efforts, this dissertation study seeks to provide a coherent and systematic multimodal data analytics and fusion framework for efficiently and effectively managing disaster-related information. Specifically, it addresses several potential challenges in developing a multimodal data analysis framework for DS, which are summarized below.

Data-Driven Knowledge Discovery: In order to provide a proper data-driven approach that can effectively address critical problems, it is necessary to collect raw data, develop suitable models, and use those models to generate insights, decisions, and policies. In other words, we must take disorganized sets of information and turn them into knowledge human beings can understand. Working with larger datasets of a higher quality can aid in this process, but doing so can also significantly increase the computational complexity without yielding improved results. Time-sensitive information, such as crisis-related data, may become worthless shortly at its creation, but the ability to extract and manage such information can benefit society by enabling data-driven decision-making and rehabilitation efforts [1, 2]. As shown in Figure 1.1, traditional data analytics uses hand-crafted (low-level) features on simple trainable classifiers in various application domains. Those diversified representatives are then combined into a single form and stored for future content analysis [3, 4, 5]. Recent advances in technology have also made it possible to record multimedia data in higher resolutions, which is a double-edged sword as it distinctively improves the analytical results by increasing the feature quality while at the same time slows the analysis process due to the exponentially increased number of features involved.

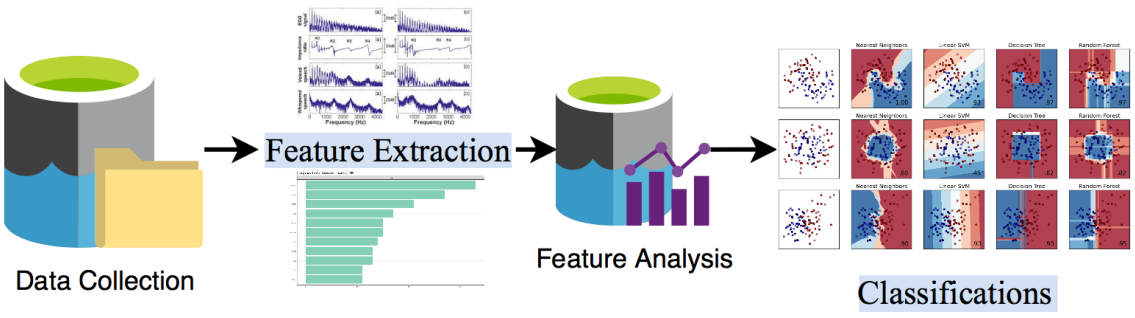


Figure 1.1: Traditional data analytics

Varied Reliability of Single Modalities: As we have learned from research on the human brain, thousands of cortexes serving different purposes activate or deactivate given

an external stimulus. However, each cortex only works with restricted signals: cortical cells respond to lines in specific orientations, while the auditory cortex wakes up when a particular frequency is received.

For example, in Figure 1.2, there are two video clips containing different semantic concepts. The upper clip records a briefing session discussing a current disaster situation while the lower one shows a flooded street with nature sounds, such as water flowing and wind blowing. Individuals with non-functioning visual or auditory cortices find ways of coping with their disabilities and leading full lives, but they inevitably miss out on important stimuli that most would prefer to experience. Restricting out analysis of data to single modalities of data when multimodal data is available is akin to artificially imposing a disability upon ourselves. It is more challenging to extract enough information from a scene using auditory stimuli. In some cases, having only one modality to react requires significant change before we notice it. For instance, by listening to the dialog in the conference room we may immediately realize that it is a briefing and identify the topic, but the sound of flowing wind and water alone may not make the flooding conditions immediately apparent.

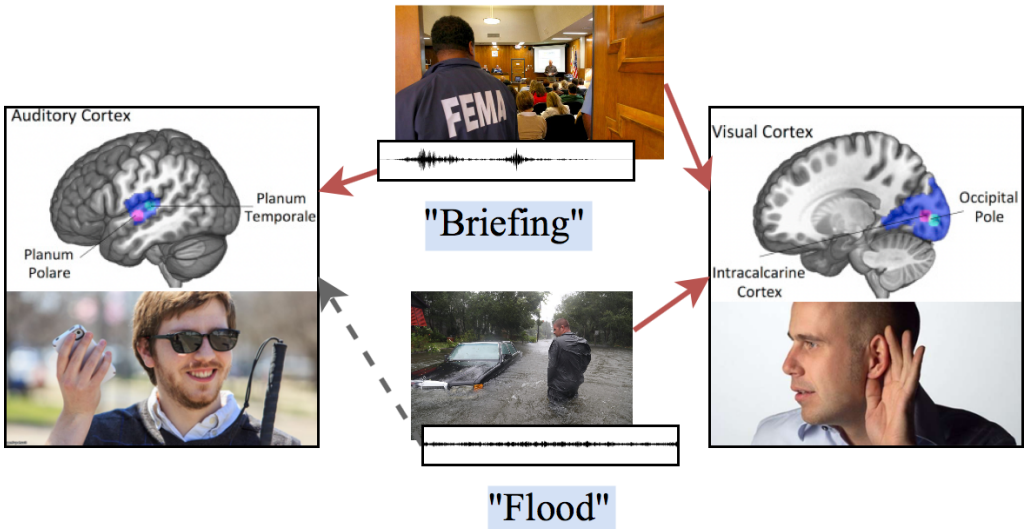


Figure 1.2: An example illustrates when the auditory cortex wakes up

In a multimedia system, using various data types can significantly improve the final detection and retrieval performance, especially when there are errors or missing values in one or more modalities. In nature, a human brain can glean concepts from a video not only by visualizing the spatiotemporal data but also by listening to the audio and reading its description. Despite the multimodality of data, traditional Machine Learning (ML) and data mining studies mainly focused on a single modality, usually textual information [6]. However, elements in Web data such as typing errors, special characters, and abbreviations may cause difficulty in semantic detection and information understanding [7]. Therefore, the need for multimodal data analysis has become apparent. As multimodal data generation and collection grows, more reliable and cutting-edge techniques are required to reap the benefits of obtaining new knowledge.

Cross-modal Semantic Gaps: More recently, multimodal deep learning techniques have been introduced to enhance the performance of deep models that focus solely on a single modal data type. By distributing tasks to each model, a multimodal framework gains the ability to handle multiple data sources and take the data analysis tasks to the next level. This type of integration framework establishes astounding performance for specific modalities and aggregates the outcomes to provide high-level semantic concepts. Although each data modality has its strengths and associated deep learning approaches, there are still some limitations. For example, when considering multiple modal inputs, the mixed semantic meanings might confuse the computer model when detecting and classifying complex semantics. Traditional data fusion techniques usually include early fusion, late fusion, and middle fusion. Contrary to early fusion, which learns the shared representation from different modalities before using a single classifier to handle cross-modal features, late fusion, also known as decision fusion, integrates the prediction results from several classifiers to generates a comprehensive result. In this case, each classifier only processes the features representing a particular modality. Both early and late fusion

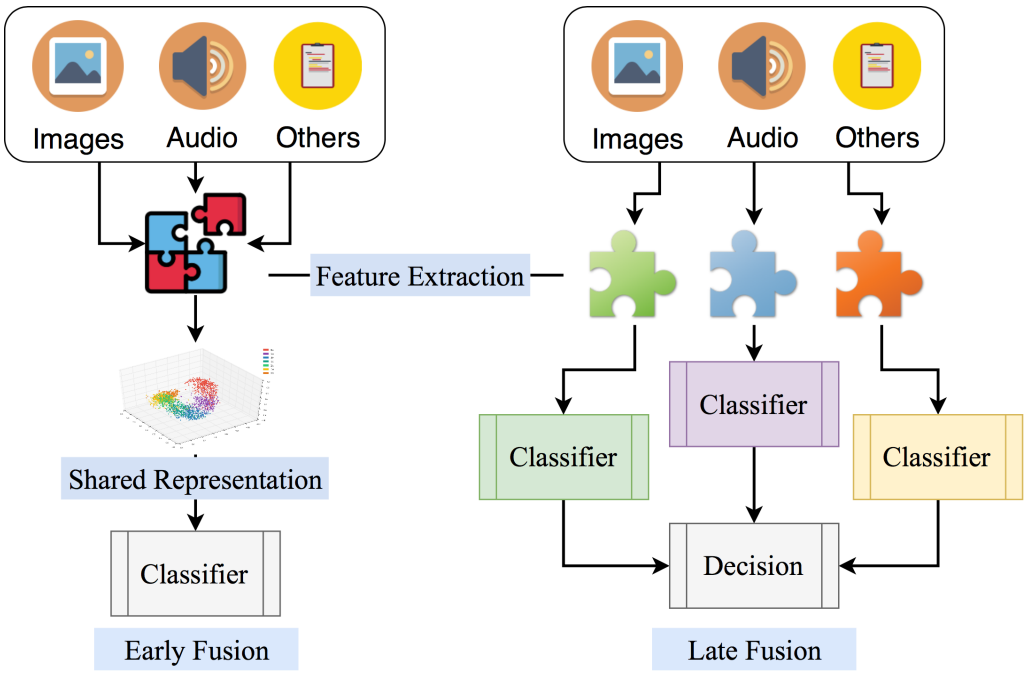


Figure 1.3: Early and late fusion strategies apply to multimodal fusion approaches

strategies can be utilized towards multimodal fusion as depicted in Figure 1.3. Topics of existing studies include, but are not limited to, video (audio-visual) analysis [8], biomedical and healthcare [9, 10], social networks [11], and human-computer interaction [12].

1.2 Proposed Solutions

In this dissertation, a systematic and integrated framework is presented to solve the problems described above. Without loss of generality, a domain-specific (i.e., disaster) dataset is used as a test bed for evaluating the major components of the proposed framework. The analytical procedures are detailed below.

Data Analysis: Various approaches have been developed to convert low-level features into high-level semantic concepts, including feature selection [13, 14], feature extraction [15, 16], and classifier selection [17, 18]. Feature selection reduces the dimensionality of the feature space in order to efficiently speed up the learning process without

compromising the quality of the results. Multilayer Perceptron (MLP) neural networks are the basis of deep learning architectures. These architectures provide a complex function to determine the feature values in the feedforward direction. However, there is still ample room for improvement. Compared to deep learning models, shallow learning models memorize rather than understand features. Many ML and data mining approaches seek to understand the precious information in the raw data, while other methods attempt to fill the gap between low-level features and high-level semantic concepts. Beyond shallow learning methods, deep neural networks like stacked MLPs target complex learning tasks in order to understand the data in greater detail. In a recent study, Genetic Convolutional Neural Network [19] was proposed to learn the structure of deep neural networks automatically using a Genetic Algorithm (GA). To that end, the authors introduced new encoding scheme that uses a fixed-length binary sequence to indicate the network structure. Finally, the F1 score on a reference set is used as a fitness function to determine the quality of each individual in a population.

Multimodal Deep Representation Learning: Real-world applications usually involve data with various modalities, each containing valuable information. In order to enhance the performance of these applications, it is essential to adequately analyze all of the information extracted from the different data modalities. However, most of the existing learning models ignore some data types focusing instead on a single modality. Recent advances in multimedia research have sparked interest in improving the detection and classification of data in closely related modalities. Early attempts to classify human actions in videos utilized spatial and temporal features procured using detectors and descriptors, which were later processed through a bag-of-features approach using Support Vector Machines [20]. A recently proposed novel approach [21] leverages the advantages of a hybrid framework that learns features from both static data (images) and optical flows. Automatically learning the structure of neural networks has likewise been

studied for many years [22, 23]. Researchers have paid significant attention to GA-based approaches to tune the network structure. In the model proposed by Ijjina *et al.* [24], a GA is used to determine the optimal weight initializations of deep neural networks. Specifically, this approach applied to a Convolutional Neural Network (CNN) to recognize human actions and avoid getting stuck in a locally optimal solution.

Semantic Concept Detection and Multimodal Fusion: Decision fusion is commonly used at the last stage before generating conclusive classification results from different classifiers. Non-linearly weighted summation is a popular methodology for exploring the interdependencies among multiple classifiers. Decision fusion schemes are widely employed to improve the performance in multimodal, multi-temporal, and multi-spatial feature classification problems. In a multimodal data analysis framework, it is important to efficiently leverage the learned feature representations from different modalities, in order to achieve maximum performance and harvest relevant information.

1.3 Contributions

The major contributions of this dissertation are the following:

- A Feature Affinity-based Multiple Correspondence Analysis and Decision Fusion (FA-MCADF) framework is proposed to extract useful semantics from a disaster dataset. The proposed framework achieves improved concept detection results by utilizing the selected features and their affinities/ranks in each of the feature groups. Moreover, the decision fusion scheme further improves the accuracy performance.
- A framework of Multiple Correspondence Analysis-based Neural Network (MCA-NN) is presented to address the challenges in shallow learning. This framework integrates the Feature Affinity-based Multiple Correspondence Analysis (FA-MCA) models into one large neural network model. The proposed semantic concept detec-

tion framework is used in place of frame-based classification in order to determine the video concept. Furthermore, the process of deciding the neural network module is automatic. The most important parameters for building the network are obtained from the output of the FA-MCA models and the corresponding statistical information.

- A new genetic algorithm for deep learning optimization and model selection is proposed. Specifically, the proposed genetic encoding can automatically select the best deep feature model from the population. Instead of manually defining an adaptive network that considers many characteristics of the datasets, the framework integrates Evolutionary Algorithms (EA) and other techniques to support the automated searching process. The hyperparameters of a new neural network for one specific task are determined after the best individual is selected.
- A multimodal deep learning framework that incorporates sequential information from both audio and textual models is proposed to assist the disaster-related video classification. For the audio model, an effective and efficient deep learning model is utilized to extract the most discriminative and high-level feature representations that is extended through a time distributed fully connected layer and the subsequent Long-Short-Term-Memory (LSTM) layers. For the textual model, a pre-trained word embedding layer is used with a stacked LSTM model to generate the video-level concepts and a novel two-stage fusion technique is proposed based on the frame-level image, audio, and video-level information by building a CNN model. Most notably, the image model predictions are incorporated into the audio model to adjust the classification ranking scores based on the reliability of the different predicted sound classes.
- A multimodal deep learning framework is proposed that utilizes different sources of information including visual, audio, and textual data. Unlike conventional fusion

techniques such as early and late fusion, a two-stage modality fusion approach is proposed to first analyze the temporal information from both visual and audio data and then combine the textual information with the results from the first stage.

1.4 Scope and Limitations

The proposed framework has the following assumptions and limitations:

- Some of the parameters are determined empirically, such as the learning rate that affects the weight updates during backpropagation in the MCA-NN algorithm.
- The proposed framework specifically focuses on improving the performance of semantic concept detection on multimedia data. It is necessary to further expand the proposed ideas into broader research topics in other domains of data analytics.

1.5 Outline

The organization of this dissertation is as follows: Chapter 2 presents the literature review in the areas of feature analysis, multimodal deep representation learning, semantic concept detection and multimodal fusion. Chapter 3 provides an overview of the proposed multimodal data analytics and fusion framework. Each component of the framework are introduced in details. Chapter 4 discusses semantic data representation solutions, especially the feature analysis method. Chapter 5 presents the deep learning approaches for semantic concept detection crossing multiple modalities. Chapter 6 introduces the proposed fusion approaches based on the statistic distributions and multimodal characteristics. Finally Chapter 7 provides conclusions along with proposed future work.

CHAPTER 2

RELATED WORK

In this chapter, the related work in the areas of data analysis, multimodal deep representation learning, and multimodal fusion for semantic concept detection will be reviewed.

2.1 Feature Analysis

2.1.1 Low-Level Feature Correlation Analysis

Multimedia data analysis has been widely used in a variety of application domains that need to process and manage huge amounts of raw multimedia data, typically represented by a group of low-level features [3, 4, 5, 25, 26]. The low-level features are image descriptors of the visual properties that are extracted directly from the images without any object description [27, 28]. The features are converged into a single form for the sacks of storage with diversified representatives and can assist the content analysis afterward. On the other hand, high-level features or concepts that contain the semantic information can be acquired from the low-level features using some data analytic approaches. In order to utilize these low-level features to characterize high-level semantic concepts, various approaches have been developed, including feature selection [14, 29, 30], classifier selection [18, 31, 32, 33], and decision fusion [34, 35].

Thanks to the technological advances that greatly enhance the quality of the recorded multimedia data, higher resolution data is widely used to further improve the analysis outcomes. However, the more features learned from the data, the more computational time it will need, which slows the analysis process. For most of the multimedia applications, especially in the current big data era, the dimension of the features is very high and

thus feature selection is commonly applied to reduce the feature dimension to make the learning more efficient [36, 37].

After the feature selection step, many ML algorithms can be used to detect the high-level semantic concepts. Some examples include Artificial Neural Network (ANN), Decision Tree (DT), Support Vector Machine (SVM), and Multiple Correspondence Analysis (MCA) [38, 39, 40, 41]. MCA has been used as a classifier which calculates the correlations between the features and the classes. DTs that use information gain to generate the tree structure are another commonly used classifier. However, while building the branch for each decision direction, the features are considered independently. SVMs can bound the generalization error and build consistent estimators from the data.

2.1.2 Deep Learning

Traditional ML heavily relies on feature engineering, an approach that generates representative handcrafted features for a specific task. Given the task at hand, the feature engineering process requires the comprehensive domain knowledge to transform raw data into valuable features. The algorithm will then take those generated features to build models that can differentiate the observations into distinct concepts. The modern world has developed new ways to consistently collect and store vast amounts of data. However, when relying on feature engineer, a major difficulty is how to find the most representative features given most of the data collected is unstructured. Therefore, multimedia data which is the most significant source of unstructured data, requires the use of more advanced artificial intelligence techniques. For instance, when it comes to processing heterogenous data such as images and video, a lot of time is consumed by the feature engineering process to reduce the dimensionality of the data by identifying a cohesive subset of attributes that best represent the data. In essence, deep learning is a new technique that has

proven to be appropriate in advancing the field of AI. It has considerably simplified the modeling process by incorporating both feature engineering and conceptual learning to directly process raw data then generate the final, conclusive results [42]. Studies from different research fields have shown how deep learning eases the research work by requiring less task-specific manual process. Some notable frameworks that have leveraged deep learning into real-world applications include recommender systems [42], answer selection [43, 44], and medical image analysis [45]. Compared to the traditional independent feature engineering effort, deep learning models have better capability to generalize unseen combinations of features by embedding sparse inputs when solving large-scale regression and classification problems.

In recent years, several extensions of successful deep learning models are introduced such as ResNetXT [46], Inception-v3 [47], and Inception-ResNet [48]. These models and their pre-trained weights on very large-scale datasets (e.g., ImageNet) have been widely utilized in different research and applications. More specifically, recent studies have shown the importance of the deep features extracted from the pre-trained models using transfer learning over traditional handcrafted features [49, 50].

2.1.3 Transfer Learning

Based on the No Free Lunch Theorems [51], there is no single form of machine learning approach could solve all the problems. Thus, a large group of deep learning architectures has been successfully developed to fit appropriately with commonly used datasets. Generally, CNNs are designed following a hierarchical architecture that consists of both linear and non-linear layers. Primarily, CNNs were intended to be utilized for basic image recognition, which made them standout amongst the most well-known and broadly utilized deep learning methods. Different from traditional Artificial Neural Network (ANN)

models such as Multiple Layer Perceptrons (MLPs), which isolate the feature layers completely, CNN models take a raw image as input with a two-dimensional structure and share the feature weights among local neuron connections. This change significantly reduced the number of parameters and made the model simpler and easier to learn.

Many CNN models are built and trained on ImageNet, a large scale public image dataset, and can be utilized in transfer learning to tackle visual data classification tasks in a broader target domain. For this purpose, instead of training an entire CNN model from scratch, many researchers run the pre-trained reference models as the feature extractors to construct new feature sets. This process is called transfer learning, and the reference models are pre-trained on very large-scale datasets, such as ImageNet.

- **InceptionV3 [47];** is an updated version of GoogleNet, which introduced a deeper and wider network [48, 52]. It is the first model that has the convolutional and pooling layers separated in parallel. Altogether, it consists of twenty-two layers of a deep system, which conserve both power and memory through the use of extra sparse layers. The main piece of this network is identified as "Inception" which generates more optimal locality and repeats it spatially. Since only a small number of neurons are effective, the width/number of the convolutional filters of a particular kernel size is kept small. Also, it uses convolutions of different sizes to capture the details at varied scales (5×5, 3×3, 1×1). The module also has a bottleneck layer (the 1×1 convolutions). This is beneficial since it aids in massive reduction of the computation requirement.
- **Residual Networks (ResNet) [53]:** Generally, ResNet overcomes the potential overfitting and vanishing gradient issue by constructing residual modules, which increase the depth of the model. Similar to InceptionV3, it uses a global average pooling followed by the classification layer.Through the changes mentioned, ResNets were learned with network depth of as large as 152.

- **MobileNet:** MobileNet [54] is an efficient lightweight CNN model for mobile and embedded vision applications. The standard convolutions are factorized into pointwise convolutions and depthwise convolutions. The core layer of MobileNet is depthwise separable filters, named as Depthwise Separable Convolution [54]. The network structure is another factor to boost the performance. Finally, the width and resolution can be tuned to tradeoff between latency and accuracy. Depthwise separable convolutions which are a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. In MobileNet, the depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1×1 convolution to combine the outputs the depthwise convolution.

- **DenseNet:** Proposed by Huang *et al.* in 2016 [55], DenseNet built the network structure which connects every layer to every other layer in a feedforward fashion. This modification obtains significant improvement by strengthening the feature propagation and encouraging the reuse of feature, which substantially reduce the number of parameters.

- **VGG16:** This architecture is from the VGG group in Oxford [56]. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layers, respectively) with multiple 3×3 kernel-sized filters one after another. With a given receptive field (the effective area size of the input image on which output depends), multiple stacked smaller size kernel is better than the one with a larger size kernel because multiple non-linear layers increases the depth of the network which enables it to learn more complex features, and that too at a lower cost. There are blocks with same filter size applied multiple times to extract more complex and representative features. This concept of blocks/modules became a common theme in the networks after VGG. The VGG

convolutional layers are followed by 3 fully connected layers. The width of the network starts at a small value of 64 and increases by a factor of 2 after every sub-sampling/pooling layer. While VGG achieves a phenomenal accuracy on ImageNet dataset, its deployment on even the most modest sized GPUs is a problem because of huge computational requirements, both in terms of memory and time. It becomes inefficient due to large width of convolutional layers.

2.1.4 Search and Optimization Algorithms

Both genetic algorithms and evolutionary programming are population-based optimization algorithms that incorporate many biological evolution operations to improve the quality of the solutions iteratively [57]. The operations include reproduction, mutation, recombination (a.k.a. crossover), and selection. A fitness function is defined to evaluate the health of each individual during the evolution process. Generally, a genetic algorithm is used to find precise solutions to both optimization and search problems, including finding either the minimum or the maximum function [58]. Compared to traditional methods, a genetic algorithm progresses from a population of candidate solutions, hence minimizing the chances of finding a local optimum. They can function under a noisy, nonlinear space, and are flexible to adjust. Recently, researchers have seen working on ways in which genetic algorithms can be used with evolutionary computation such as neural networks. Evolutionary programming is used in evolution simulation and to maximize the suitability of multiple solutions within an objective function. It relies on a known gradient within the search space when applied to design problems whose objective is the creation of new entities [59]. The recombination operation is eliminated from evolutionary programming because it considers each individual as an independent species. However, its advantage is the same as that of genetic algorithms, where no assumption is made about

the underlying fitness landscape. Compared to other methods, they perform well on approximating solutions for nearly all types of problems and act efficiently when combined with neural networks.

Grid search is used to perform hyperparameter tuning to determine the optimal value for a specific model. Compared to genetic algorithms, grid search helps to find near-optimal parameter combination within specified ranges, such as support vector machine parameter optimization [60]. Gradient-based optimization can be applied to the optimization of neural network's learning rate separately for every iteration and layer. Compared to manual tuning, it enhances the ability to learn completely new data sets. However, the main disadvantage is that backpropagation across the entire training procedure requires a lot of time.

Random search algorithms are used to randomly select a representative samples from a given search space in order to identify the optimal value in the sampling [61, 62]. It does not require derivatives to search in a continuous domain. Compared to grid search, the chances of finding optimal parameters are higher because of the random search pattern. Random search is faster than exhaustive search, but it is unreliable in determining the optimal solution.

A Bayesian optimization algorithm is a powerful tool when it comes to joint optimization design choices due to its ability to increase both product quality and productivity of human beings through an enhanced automation capacity [63]. It has been popularly used in many application domains, including interactive user-interfaces, environmental monitoring, automatic network architecture configuration, and reinforcement learning. Primarily in reinforcement learning, Bayesian optimization is used to tune the parameters of a neural network policy automatically, and to learn value functions at advanced levels of the reinforcement learning hierarchy. The technology can also be used to determine attention policies within image tracking with the use of deep neural networks. Compared to man-

ual tuning methods, this approach can be used to tune many parameters simultaneously, which is essential for machine learning systems. The disadvantage with this technology, however, is that it is independent and relies on an optimizer to search the surrogate surface. Different from the general problem domains that we have observed, Bayesian optimization attains a superior performance, the relationship between each layer's feature performance for a specific CNN model is unknown. Since Bayesian optimization assumes that the solution space reflects the posterior probability distribution, it is uncertain if it is a good fit of Bayesian optimization for deep learning model selection.

2.1.5 Automated Neural Network Construction

Many existing deep learning models have been successfully applied for different tasks. However, an automated approach to select the best model for each dataset and each domain is not available. To address this challenge, Long *et al.* [64] introduced Joint Adaptation Networks (JAN) that is based on a Joint Maximum Mean Discrepancy (JMMD) criterion to learn a transfer network by aligning multiple domain-specific layers (layer $fc7$ in AlexNet and layer $pool5$ in ResNet).

In [65], the authors proposed a Genetic Algorithm (GA) approach using transfer learning to enhance the performance of the CNN model in the image classification tasks. Deep features were generated from four pre-trained CNN models, which are ResNet50, Inception-v3, VGG16, and MobileNet. The experimental results showed that the proposed GA method can improve the performance of the baselines. However, while a straightforward genetic algorithm method can select the primary data representation model, it needs to be extended to enable deep neural network construction for specific tasks. Moreover, genetic algorithms will not scale here, or in natural evolution. What is needed are heuristic accelerators. Such heuristics can be learned and applied in a network con-

figuration of neural networks. This provides coherency, a guiding necessary AI principle, and self-reference. The latter provides us with insight. Just as one of AIs failings led to the field of ML, so too does the failing of deep learning lead to the need for heuristics and heuristic acquisition. To fix the architecture of a hidden-layer neural network is to unnecessarily restrict that, which can and needs to be learned. Furthermore, it is argued that neural-based symbolic representations need to be enabled. It is well-known that modus ponens cannot be achieved without a symbolic representation. The creation of heuristics and their transfer-extension follows suit. The over-arching implication here is that today's deep learning architectures are not of sufficient Kolmogorov complexity to hold and learn to generalize strong knowledge. Both of these capabilities are inherent to not only real-world functionality, but commonsense reasoning as well. Commonsense reasoning has evaded capture by symbolic and neural AI alike. These are complex concepts; and, it will take some time to realize them in practice.

Automatically learning the structure of neural networks has been studied for many years [22, 66]. Many researchers have utilized the GA-based approaches to tune the network structure. Specifically, Leung *et al.* [67] proposed a method to handle both network structure and its parameters simultaneously. In that work, many network parameters were selected manually or fixed to a specific number due to the high computation costs of GA and hardware limitation. Tsai *et al.* [22], on the other hand, proposed a more robust method using the Hybrid Taguchi-Genetic Algorithm (HTGA) to enhance the traditional GA for better and faster convergence. The authors in [62, 68] discussed the advances in image classification with hyper-optimization. Computer clusters with large processing capacity GPUs allow trails and tests to be run. The researchers used hyper-optimization for training neural networks and deep belief networks, by optimizing hyperparameters with random searches and two greedy sequential methods. Sequential algorithms were applied to complex deep belief learning problems and improved results were obtained.

The researchers validated the Gaussian Process Analysis (GPA) approach with a random sampling of the Boston housing data for a regression task. The dataset has 13 scaled input variables composed by 506 points to obtain a scalar regression output. An MLP network was trained with 10 hyperparameters. The hyperparameters included the hidden layer size, learning rate, iteration times, the Principal Component Analysis (PCA) preprocessing, and others. Sampling was used for the first 30 iterations, differentiated random samples were used for training, and the whole set up had 20 repetitions. Five GPUs were used; and, the test was run for 24 hours. The results would help other researchers to develop ML with hyper optimization and genetic algorithm.

Recent research focuses on evolving the deep neural networks parameters or structures with GAs [69, 70]. In [71], an improved genetic algorithm was proposed to tune the structure and parameters of a 3-layer FFNet. Unlike deep neural networks that contain more complex structures, this network has a relatively simple structure which contains only one hidden layer. Therefore, there were few combinations of the available hyperparameters. So the best choice can be easily identified in advance.

In recent years, by the advent of deep learning algorithms, researchers have studied the possibility of learning parameters [69, 72], network structures [73], and hyperparameters [23] in deep neural networks using the GA algorithms. Young *et al.* [23] proposed a method called Multi-Evolutionary Neural Networks for Deep Learning (MENNDL) to optimize hyper-parameters in CNNs using GA. The fitness function used in that work is simply the testing error on the dataset after a specific number of iterations. The hyperparameters include the kernel size and the number of filters in each CNN layer.

In another work, GA algorithm was used to optimize the parameters in Deep Belief Neural Networks (DBNN) for object recognition [70]. In particular, parameters such as the number of epochs, learning rates, and hidden units in DBNN are optimized to decrease the training time and error rate of the object recognition task. In the work proposed by

Ijjina *et al.* [24], GA was used to determine the optimum weight initializations of deep neural networks. Specifically, it was applied to a CNN classifier for the task of human action recognition in order to avoid getting stuck in a local optimum solution. In a recent work, Genetic CNNs [73] were proposed to learn the structure of deep neural networks automatically. The suggestion is to use GAs, since the network structures tend to rise exponentially with the number of layers. To serve this purpose, a new encoding scheme was suggested, which used a fixed-length binary sequence to indicate the network structure. Then, the accuracy on a reference set was used as the fitness function to determine the quality of each individual in a population. For each generation, the standard genetic operations were defined and these include the crossover and selection mutation needed to develop outstanding individuals while rejecting weaker ones. A standalone training method was used to identify the competitiveness. The genetic process was carried out on CIFAR10 with a small dataset to examine the capability to identify high quality structures. The output of the learned powerful structures was transferred to the ILSVRC2012 data that can be used for large visual recognition.

An alternative method of hyperparameter optimization for deep neural networks is presented in [74]. It compares the proposed approach, named Covariance Matrix Adaptation Evolution Strategy (CMS-ES), with the state-of-the-art Bayesian optimization algorithms for tuning hyperparameters of a CNN network. In their work, only two optimizers such as Adam and AdaDelt can be selected, which makes the expected performance more narrow.

2.2 Multimodal Deep Representation Learning

2.2.1 Neural Networks

Essentially, ANNs are inspired by the behavior of different types of neurons in a biological system. A group of neurons that share the same properties will be responsible for the tasks related to a certain level, for example, detecting bright colors. The first level neurons' outputs will become a collection of inputs for the next level's neurons. ANNs can learn and recognize the observed patterns from this procedure. The first and simplest development in ANN is the Feed-Forward Neural Network (FFNet). It is described as a collection of associated neurons with comparative properties of the neural system located in an aminal's brain. which is a set of inter-connected neurons with similar property of the neural structure found in an animal's brain. FFNet provides the capacity for every neuron to receive signals, process the signals, and also send the accompanying output signals. For every neuron linking, there is a load factor to demonstrate the significance of the neural links. Since it is based on feed forward, the data transferred between the neurons only move towards one direction. Multi-Layer Perceptron (MLP) was introduced to address the challenge of classifying nonlinearly separable inputs [75]. In an MLP, neurons are placed in a network of multiple layers - one input layer, multiple hidden layers, and one output layer. The main objective for the hidden layers is to modify the input in a format that the output layer can use. The MLPs are used as the base of the deep learning architectures, which provide a complex function to determine the feature values in the feedforward direction. Deep learning refers to the learning process that involves more than one non-linear feature transformation step [76]. Along with transforming the low-level features into mid-level and high-level features, the level of abstraction increases with the hierarchical representations. There is no clear differentiation between recently favored deep learning networks and traditional MLP networks. However, deep learning models

usually consist of many types of layers designed for different purposes, such as reducing the feature space, obtaining the temporal information from sequences, and learning the spatial relationship between the pixel values.

Convolutional Neural Network

With the emergence of deep neural networks, we have witnessed a revolution in many area such as computer vision [77], Natural Language Processing (NLP) [78], and speech/audio processing [79]. Specifically, CNNs has shown notable improvements in visual data analytics such as image classification [80], object detection [81], and video event detection [49].

CNNs has a hierarchical structure consisting a cascade of linear and non-linear layers. It is originally proposed by LeCun *et al.* [82, 83] for simple image recognition and becomes one of the most popular and widely used deep learning techniques. To fully utilize the two-dimensional structure of an input data (e.g., image signal), local connections and shared weights in the network are utilized, instead of the traditional fully connected networks (a.k.a. MLPs). This process results in fewer parameters, making the network much faster and easier to train. This operation is similar to the one in the visual cortex cells of cat's brain. These cells are sensitive to small sections of a scene rather than the whole scene. In other words, the cells operate as local filters over the input and extract spatially-local correlation existed in the data. That version of CNNs (LeNet-5) consists of two convolutional layers each followed by a subsampling layer and finally ended with a fully connected layer for class prediction. Later on by the progress of hardware technology (e.g., GPUs), it has been widely used in many research and real-world applications [84, 85, 86].

In 2012, AlexNet [87] is proposed for image classification which extends the traditional CNNs and could achieve the best results in ILSVRC 2012 by more than 10% im-

provement in the top 5 test error. This model utilizes the GPU implementation of CNNs together with image augmentation and dropout techniques to handle overfitting problem. After that, a surge of research studies has been started to investigate the capability of CNNs in visual data analytics. Some studies mainly focus on the new structures of deep networks by introducing deeper [53, 56] and wider [88] CNNs. Both VGGNet [56] and GoogleNet [88] are presented in ILSVRC 2014 and introduced very deep CNNs to further improve the image classification results. VGGNet, particulary, proposes a very simple model with 19 CNN layers, while GoogleNet, the winner of ILSVRC 2014, introduces a more complex module (Inception) which applies several operations such as convolution and pooling in parallel. In 2015, ResNet [53] is proposed by Microsoft Research and achieve remarkable results in ILSVRC and COCO 2015. This model introduces residual connections in CNNs to overcome overfitting in very deep networks. This results in designing an ultra deep CNN with more than 100 layers.

Recurrent Neural Network

Another widely used and popular algorithm in deep learning, especially in NLP and speech processing, is Recurrent Neural Network (RNN) [89]. Unlike traditional neural networks, RNN utilizes the sequential information in the network. This property is essential in many applications where the embedded structure in the data sequence conveys useful knowledge. For example, to understand a word in a sentence, it is necessary to know the context. Therefore, an RNN can be seen as short term memory units which include the input layer x, hidden (state) layer s, and output layer y.

In [90], three deep RNN approaches including deep "Input-to-Hidden", "Hidden-to-Output", and "Hidden-to-Hidden" are introduced. Based on these three solutions, a deep RNN is proposed which not only takes advantage of a deeper RNN, but also reduces the difficult learning in deep networks.

One main issue of an RNN is its sensitivity to the vanishing and exploding gradients [91]. In other words, the gradients might decay or explode exponentially due to the multiplications of lots of small or big derivatives during the training. This sensitivity reduces over time, which means the network forgets the initial inputs with the entrance of the new ones. Therefore, Long Short-Term Memory (LSTM) [92] is utilized to handle this issue by providing memory blocks in its recurrent connections. Each memory block includes memory cells which store the network temporal states. Moreover, it includes gated units to control the information flow. Furthermore, residual connections in very deep networks [53] can alleviate the vanishing gradient issue significantly.

Compared with the traditional RNN layer, it is proved that the gated units, both the LSTM unit and GRU, show the evident superiority in more challenging tasks, such as raw speech signal modeling. The key difference between them is that a GRU has two gates (i.e., the reset and update gates); whereas an LSTM has three gates, namely the input, output, and forget gates. In the LSTM unit, the amount of the memory content in the network is controlled by the output gate. On the contrary, the GRU unit controls the flow of information like the LSTM unit, but without having to use a separate memory cell. It just exposes the full hidden content without any control. The GRU controls the information flow from the previous activation when computing the new candidate activation, but the control is tied via the update gate.

2.2.2 Multimodal Representation Learning

The advances in multimedia research have sparked the interests in improving the detection and classification from data in closely related modalities. Video classification has been positively impacted by the improvements in the detection and classification of objects within images [28, 41, 93, 94, 95, 96, 97, 98]. Early approaches to classify human

actions in videos utilized spatial and temporal features procured using detectors and descriptors that are later processed through a bag-of-features approach using SVMs [20]. In [99], deep learning techniques were introduced to build a model that learns invariant features from spatio-temporal data. Video classification is nowhere near the stage and scale of image classification in the multimedia data mining field. In [100], CNNs, having the best advantage in image classification tasks, are proposed to classify videos from a dataset comprised of over 480 sports videos. In contrast to CNNs, RNNs show promising performance in handling and modeling temporal and/or sequential behavior. Among the most frequently used RNN models, the LSTM networks have shown its promise in speech recognition, language modeling, and more generally, any classification or prediction task where the problem has sequential or temporal traits.

The introduction of multimodal deep learning techniques enables a significant improvement compared with using a single modality alone. This motivates the researchers to build deep networks that could learn, improve, and fuse knowledge in order to achieve a higher prediction accuracy when different modalities (image, audio, text, etc.) share similar semantic concepts. Recently, a novel approach proposed in [21] leverages the advantages of a hybrid framework that learns features from both static data (images) and optical flows. Multimodal deep learning approaches encompassing data modalities beyond image, audio, and video are still very few. Since text data can be obtained as easily as audio and video, it can also help to improve the accuracy in deep learning frameworks. Global Vectors for Word Representation (GloVe) [101] is a favorite technique that works with word embedding and maps text words into a real vector domain.

2.3 Multimodal Fusion for Semantic Concept Detection

Decision fusion is the last step before printing out the merged classification results [102]. It commonly uses non-linearly weighted summation methodologies to explore the interdependencies among multiple classifiers. Decision fusion frameworks are widely employed for multi-modality, multi-temporal, and/or multi-spatial feature classification problems.

Recently, social media and Web have become some of the most significant sources of information for various events. To understand the user behavior on the Web, many research studies have analyzed the user-generated data [103]. Despite the multimodality nature of these data, traditional studies mainly focused on a single modality especially textual information [6, 104]. For instance, semantic evaluation is an important field to understand the human thoughts and opinions through the Web and social media conversations [105]. However, there are several challenges such as typing errors, special characters, and abbreviations which may cause difficulty in semantic detection and information retrieval from Web data [7]. Therefore, the need for multimodal data analysis has become apparent.

In multimedia systems, it is important how to integrate different modalities from the data in order to achieve maximum performance and harvest relevant information [106, 102]. Traditional data fusion techniques usually include early fusion, late fusion, and middle fusion [107, 108]. The existing multimodal fusion studies include, but are not limited to, video (audio-visual) analysis [8], biomedical and healthcare [9, 10], social networks [11], and human-computer interaction [12]. The work in [11] presents a tri-modal sentiment analysis based on textual, video, and audio modalities. It first generates handcrafted features for all modalities and then employs a tri-modal Hidden Markov Model (HMM)-based classification to discover the hidden interaction between them.

Deep learning techniques have been recently proposed and applied in many research, competitions, and real-world applications [49, 109, 110]. Specifically, Microsoft proposed Residual Networks (ResNet) [53] which overcome the overfitting and vanishing gradients problems in very deep networks. Inception networks [48] were originally proposed by Google in 2014 for the ILSVRC ImageNet competition and later were improved and combined with ResNet to enhance the classification performance. These two models have been widely used in the literature and different visual applications for feature extraction and model fine-tuning. Other input data types such as textual and audio are also widely studied in the deep learning community. Kim [78] applied a simple CNN on top of word vectors for sentence-level classification specifically for the application of sentiment and question analysis. RNN and its extended version LSTM have also been leveraged to classify speech [111] and text [112].

All aforementioned deep learning studies have mainly focused on a single application (e.g., NLP, speech, vision). Few deep learning studies in the recent years consider different types of multimedia data such as audio, text, and image. EmoNets [113] is an example of multimodal deep learning for video emotion recognition that includes a CNN for face visual analysis, a deep belief net for audio processing, and a relational autoencoder for spatio-temporal analysis of videos. However, this model did not utilize textual information or metadata from the videos. Karpathy *et al.* [100] proposed different fusion techniques for large-scale video classification using CNNs. However, it only focused on visual information from videos.

CHAPTER 3

OVERVIEW OF THE PROPOSED FRAMEWORK

New technological advances have resulted in the generation of huge volumes of data which have had a powerful impact on various areas in science and engineering. Numerous research opportunities call for the design and development of advanced approaches in big data analytics. Given such demands, data science has become an emerging hot topic in both industry and academia, ranging from basic business solutions, technological innovations, and multidisciplinary research to political decisions, urban planning, and policy-making. However, existing tools and techniques are still far beyond satisfactory in terms of collecting, analyzing, and managing multimodal data. This dissertation proposes a multimodal data analytics and fusion framework for data science, especially for disaster-related information management. As shown in Figure 3.1, the proposed framework consists of three major components: data analysis, multimodal deep representation learning, and multimodal fusion for semantic concept detection. These three components are seamlessly integrated and act as a coherent entity to support the framework's fundamental functions. Specifically, the data analysis component represents the semantic information of each modality and serves as the basis for the other two components. The multimodal deep representation learning component evaluates and organizes the multimodal data, which helps with efficient semantic concept detection based on multiple input channels. The fusion component applies the strategies that are necessary to boost the performance of models built on the isolated data modalities.

The proposed framework will explore useful knowledge hidden in different data formats and incorporate representation learning from multimodality data. An example application for disaster information management is demonstrated. First, we present a Feature Affinity-based Multiple Correspondence Analysis (FA-MCA) method to analyze the correlations between low-level features, and we propose an MCA-based Neural Network

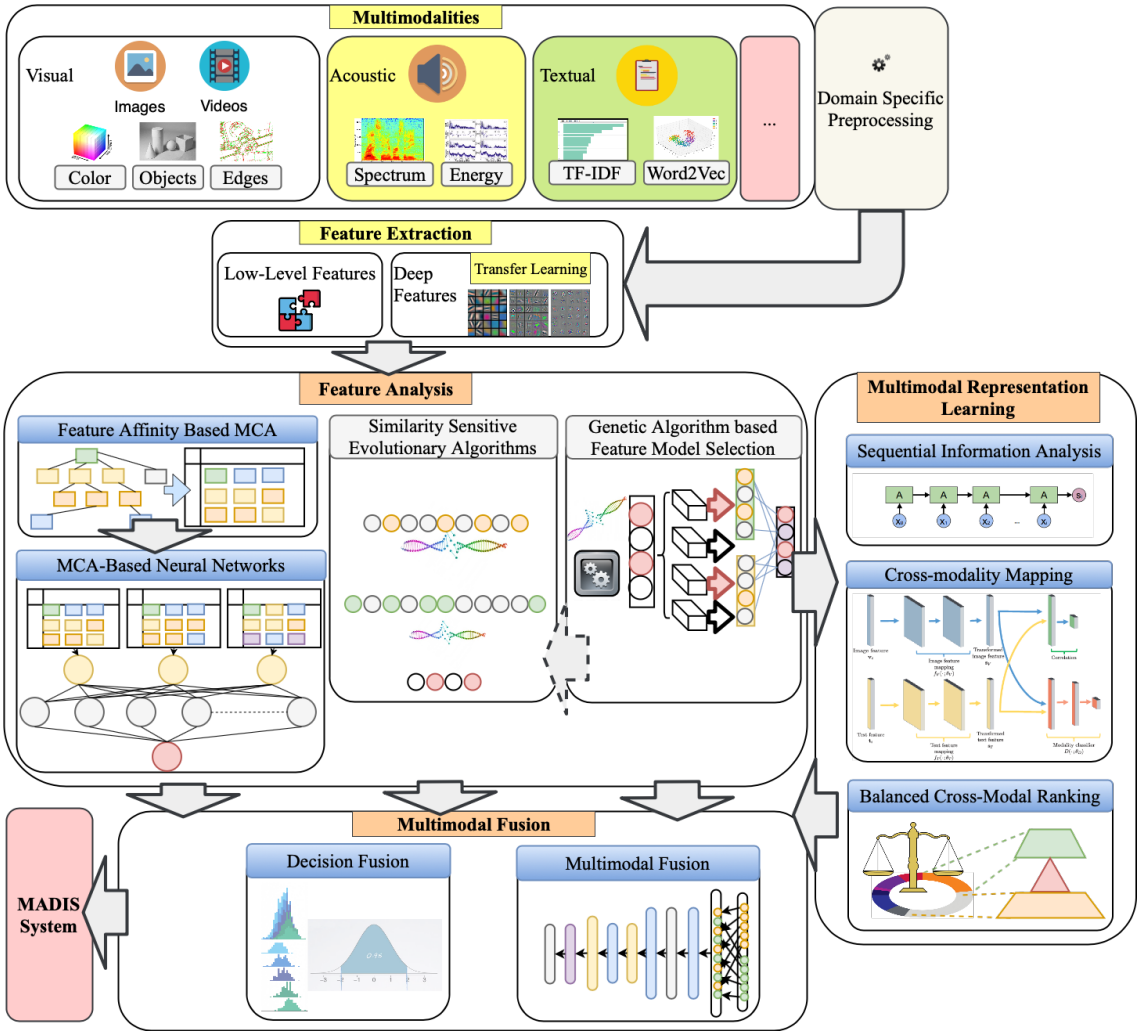


Figure 3.1: Overview of the dissertation's framework

(MCA-NN) to capture the high-level features from individual FA-MCA models and seamlessly integrate the semantic data representations for video concept detection. Next, we present a genetic algorithm-based approach for deep neural network selection. The genetic operations are integrated with deep neural networks to generate populations that can identify the optimized deep representation learning models. Then, we propose a multimodal deep representation learning framework to incorporate the semantic representations from data in multiple modalities in an efficient manner. Finally, we apply fusion strategies to accommodate multiple modalities. In this framework, we also propose

cross-modal mapping strategies to organize the features in a better structure to improve the overall performance.

3.1 Data Analysis

In this age of the Internet, people frequently interact with digital devices. These devices range from mobile phones, tablets, sensor-equipped infrastructures, vehicles, to smart household appliances. With this, we are experiencing a surge in data generation and transmission, which affects our everyday lives. More specifically, in multimedia data generation, which plays a vital role in both industrial applications and academic research [114, 115]. This generation makes up 70% of the daily generated Internet data, and these vast amounts of data can be utilized to solve various domains' problems.

Multimedia semantic concept detection has been one of the major research topics in multimedia data analysis in recent years. The widespread growth of multimedia data including video, image, audio, and text has provided extensive opportunities in various big data applications [28, 93, 96, 116]. Among them, visual data analytics is a fundamental task in multimedia data. Its applications include video event detection [49], autonomous driving [117], robotics [118], healthcare [119, 120], and disaster management [121, 122, 123]. Disaster information management needs the assistance of multimedia data analysis to better leverage disaster-related information that has been widely shared by people through the internet. Disasters can disrupt a community, causing serious human, economic, and environmental losses [1, 2]. Natural catastrophes and accidents produce a large amount of time-sensitive information [124], and having the ability to collect, analyze, and manage such information would benefit society in critical decision-making and recovery efforts [125, 126]. The recent exponential growth of multimedia data coupled with the semantic richness of visual information have made visual data a popular tool to conquer the challenges in disaster information management [126, 127, 128].

In this work, we seek to improve the concept detection results by feeding the learned results from individual shallow learning models into a generic model to uncover the similarities in deeper layers. The domain-specific preprocessing phase is normally needed to clean and better organize the data. Feature analysis is established to represent the raw data in a more structured way. Performing feature analyses, such as feature extraction and feature selection, could improve the performance within a reasonable time frame. High-level features that capture semantic meanings are transformed through a careful analysis of the low-level features. There has been much research effort to bridge the semantic gap between them [18, 41]. In this work, we utilize a dataset obtained from the Federal Emergency Management Agency (FEMA) website for detecting disaster-related semantic concepts. The semantic concepts obtained from this website are different from normal disaster event concepts. Not being restricted to the disaster classification tasks that attempt to classify the disaster scenes from non-disaster scenes means we can utilize a variety of information relevant to a specific disaster, including the hazard situation, recovery progress, disaster effects, and disaster prevention, to name a few. The difficulty increases, however, because all those concepts are surrounding one major premise, which will immensely increase the similarity between the concepts.

We propose utilizing a Feature Affinity-based Multiple Correspondence Analysis (FA-MCA) to extract useful semantics from disaster datasets. By utilizing the selected features and their affinities/ranks in each of the feature groups, the proposed approach will be able to improve concept detection results. Furthermore, to tackle the issue of shallow learning, we propose a novel framework that integrates the strengths of Multiple Correspondence Analysis (MCA) and MLP neural networks. The low-level features are the initial inputs for MCA-based models that extract higher-level features. The output of each FA-MCA model further involves interaction in the neural network for better semantic understanding, generating the ability to put forward arguments. Specifically, the proposed

framework can automatically decide the structure and the initial parameters of the neural network module. Furthermore, the model obtains the most important parameters building the network from the outputs of the FA-MCA models and the corresponding statistical information.

Nowadays, remarkable progress has been achieved in visual data analytics due to advanced machine learning and deep neural network techniques. Advanced techniques, such as Deep Learning (DL), have been popularly used to investigate the different ways to take advantage of multimedia data analysis in different research fields [129]. Many astonishing research outcomes are generated with the assistance of DL approaches, including image classification [65], speech recognition [130], video understanding, etc. However, it is time-consuming and computationally expensive for each research group to build a DL model from scratch to fulfill their targeting solution. A method to better simulate a person's learning process is needed to generalize the knowledge to help solve these problems. Transfer learning, which provides the ability to transfer the experience from an original problem domain to a target domain, eases the learning process, and makes the well designed pre-trained models useful in a broader application domain as feature extractors [131, 132].

Pre-trained deep learning models can extract different levels of features from the input data. However, for a variety of datasets, the feature strength is also varied [133]. In particular, Convolutional Neural Networks (CNNs) have been extensively used for image classification and recognition [53, 87, 88]. These accomplishments are primarily due to the powerful machines (e.g., with GPUs) and availability of large-scale annotated datasets (e.g., ImageNet). Although the existing CNN models have proven to be effective, the networks are usually designed manually for different tasks. In addition, different networks perform well for different tasks and datasets [49]. Therefore, to efficiently identify the best model to generate the most representative features for the targeting problem

domain is challenging. In the early stage, researchers always select the last layer before the prediction layer to extract the high-level abstract features for their specific tasks. It is uncertain whether this layer of every pre-trained model can always be the best choice considering the target domain is slightly different from the original problem domain. Hence, extracting features from other layers that carry lower level features might be more suitable for a particular target domain. Regarding examining a set of layers of each popular pre-trained model to obtain the best model for a specific task, we are looking for an optimal solution from a very large search space that exceeds the human ability. Therefore, an efficient and effective optimization/search algorithm is necessary to be used to automatically generate the feature set for a specific target problem. When the input dataset changes, the framework should have the ability to evaluate each model's performance regarding the new characteristics of the data, then select a new model/layer that generates the most representative feature to build the discriminative model.

Unlike existing work, in this dissertation, we propose a new genetic encoding model that studies different pre-trained models in the population. The GA model automatically selects the best model from the population or regenerates the new candidates using GA operations. It is also worth considering the possible optimization of the learning parameters, network structures, and hyperparameters in deep neural networks using GA algorithms.

3.2 Multimodal Deep Representation Learning

Thus far, deep neural networks, a major breakthrough in machine learning, have been directly utilized in many real-world applications such as autonomous vehicles, games, science, and even art [76, 134, 135]. Deep learning has led to groundbreaking advances in different fields such as computer vision [87], NLP [112] and speech processing [111]. Most of these studies addressed the problem of single modal deep learning rather than

the challenges in multimodal learning; however, in multimedia systems, using various data types can significantly improve the final detection and retrieval performance, especially where there are errors or missing values in one or a few modalities. This is how a human brain can detect events or concepts from a video by not only visualizing the spatio-temporal data but also by listening to the audio and reading its description.

Videos serve to convey complex semantic information and facilitate the understanding of new knowledge. However, when mixed semantic meanings from different modalities (i.e., image, video, text) are involved, it is more difficult for computer models to detect and classify the concepts (such as floods, storms, and animals). Because deep learning methods require many more training examples compared to traditional machine learning approaches, the existing data sources cannot provide such a large-scale dataset at this stage. We collected a new dataset that includes several complex semantic concepts related to disaster events from YouTube website to fulfill this need. We further collected four hundred Hurricane Harvey related videos with the corresponding text information by utilizing the YouTube API in a crawling process, and we processed the videos following a series of preprocessing steps, such as keyframe extraction and audio track generation and labeled each video both at the frame-level and the video-level. First, humans annotated each keyframe image. Then we created a video-level concept based on the distribution of the frame-level concepts included in the video. Additionally, reading the video description helped us in determining each video concept.

In this dissertation, we present a new multimodal deep learning approach to detecting events in videos by leveraging recent advances in deep neural networks. First, we utilize several pre-trained deep learning models to extract useful information from multiple modalities by applying recent advances in transfer learning and sequential deep learning models. Additionally, it takes into account the prediction conflicts across modalities to balance the effect of each modality for different scenarios. Because of the natural lim-

itations of each modality, and because different semantic concepts might have different dominant features, the proposed framework needs to handle the potential conflict and make the model robust regardless of the characteristics of the concepts. In contrast to existing deep learning methods, we present a new multimodal deep learning framework using three different modalities; video frames, video descriptions, and audio information. This framework leverages both spatial and temporal information from video and effectively integrates them using a two-stage fusion technique. To the best of our knowledge, this is the first multimodal deep learning framework designed for natural disaster video analysis and retrieval.

3.3 Semantic Concept Detection and Multimodal Fusion

Real-world applications usually encounter data with various modalities, each containing valuable information. To enhance these applications, it is essential to effectively analyze all information extracted from different data modalities. However, most existing learning models ignore some data types and only focus on a single modality.

Decision fusion is commonly used at the last stage before generating conclusive classification results from different classifiers. Nonlinearly weighted summation is a popular method for exploring interdependency among multiple classifiers. Decision fusion schemes are widely employed to improve the performance in multimodal, multitemporal, and multispatial feature classification problems. In a multimodal data analysis framework, it is important to integrate different modalities from the data to achieve maximum performance and discover relevant information.

We design the fusion model following a two-stage process to handle the frame-level and the video-level multimodal representations. The first stage takes the frame-level classification results as input and generates a joint representation for the visual and audio

data, mapping the frame-level classes to the video-level classes. In the second fusion stage, namely, the video-level late fusion, textual results are combined with the audio-visual results from the previous stage to generate the final video classes.

3.4 The Disaster Application Based on Proposed Framework

The Multimedia-Aided Disaster Information System (MADIS) was proposed and first implemented in 2012 [136]. It was introduced to the Miami-Dade County Emergency Office Center with the goal of assisting disaster management personnel. The system integrates the official situation report with multimedia data (images, videos, social media posts) [125], which aims to help people understand the current situation during disaster events through mobility and valuable information integration. The system processes the plain text situation report to show the most relevant multimedia data. For example, based on the locations and disaster scene extracted from the textual information, relevant videos can be listed as references to make the report understandable.

The proposed framework has the capability to integrate the advanced deep learning technology into real-world applications. By identifying an accurate relationship between the multimodality information, the situation reports can be better associated with social media information. This will help with mapping a professional document with a lot of domain-specific terms into a comprehensible visual-enabled education tool. Therefore, the proposed framework can be incorporated into MADIS to further improve the social media data retrieval related to a disastrous event.

CHAPTER 4

DATA ANALYSIS

Benefited from the enhanced quality and increased resolution of multimedia data, a large number of features can be extracted and utilized to improve the accuracy of semantic concept detection. Though feeding these features to a powerful classifier could improve the results, it may not be an optimal one and as a result, the computational complexity will increase significantly as well. In the literature, various classifiers have been used to identify the inherent concepts in videos, including ANN [137], Logistic Regression (LR) [138], DTs [38], SVMs [39], etc. Besides performing as single classifiers, SVMs are also considered as good candidates for the choice of basic classifiers that achieve multiple decision fusion tasks. However, there is still a large space for improvements.

In this chapter, we seek to improve the concept detection results by feeding the learned results from individual feature learning models into a generic model, to dig out the similarities in deeper layers. Specifically, a Feature Affinity-based Multiple Correspondence Analysis (FA-MCA) is proposed to extract useful semantics from a disaster dataset. Furthermore, a novel framework that integrates the strengths of Multiple Correspondence Analysis (MCA) and MLP neural networks is proposed to tackle efficient semantic concept detection by taking FA-MCA prediction results from each feature group in parallel as the values of neurons. GA-based approach is introduced at the end of this chapter, which opens a new branch of feature analysis that could optimize the feature evaluation process including deep features.

4.1 Feature Extraction

The preprocessing phase is typically domain-specific. For example, in video analysis, in order to perform the frame-based classification, it usually includes key frame extraction

and feature extraction, which make the data cleaned and structured. Therefore, each video is processed independently to extract several low-level features as frame-based. To reduce the redundancy of the frames in each video, the raw videos are grouped into different video shots [94, 139]. A key frame from each video shot is selected to represent the video shot, and all the selected key frames are then used to cover the general idea of the video. This can reduce the computation time significantly.

4.1.1 Keyframe Extraction

In order to perform the frame-based classification, the first step is to generate a set of keyframes for each video which represent the whole video shots. For this purpose, a shot boundary detection method [140] is applied to identify the boundary of each shot automatically. Thereafter, at least one keyframe for each shot is extracted, and multiple keyframes might be saved for one single shot to maintain as many variations as possible. Because of the editing process of the video, some contiguous frames from the same original shot are cut and re-ordered by inserting some not related shots in between. In this step, we reduce the duplication of similar shots by detecting the similarity of the keyframes in a sequence of keyframes. At the same time, if the variety of the frames in one shot is high, we will keep more keyframes for this shot since they might better present the storyline of the whole video. In addition, the concepts might be changed during the movement of the camera. Considering the nature of user-captured videos, we try to keep as much information from each one of them as possible. Normally, those videos might only contain one shot with a relatively long time duration. In that case, if the differences between the frames within one shot is significant, we decide to keep both of the images and may assign them with different concept labels in the future. Besides removing the

redundant keyframes appear in one or multiple shots, several types of noisy frames are removed when we prepare the dataset, such as blurred frames and transition frames.

4.1.2 Low-Level Feature Extraction

Visual content is a critical modality that several different types of low-level features can be extracted from the raw data, which include Histogram of Oriented Gradient (HOG) [141], Color and Edge Directivity Descriptor (CEDD) [142], Haar-like feature [143], and color space information [144, 145]. Specifically, HOG feature is used for the purpose of object detection, which is computed on a dense grid of uniformly spaced cells and uses overlapping normalization for accuracy improvement. CEDD feature, as it is named, obtains color information and texture information. The haar-like feature is always used in object recognition with Haar wavelets, especially useful in face detection. The color space representations are the Hue, Saturation, and Value (HSV) with YCbCr as the supplemental information. As a result, one video is represented by several key frames, and each key frame is composed by several feature values.

4.1.3 Deep Feature Extraction

Deep features are extracted from several pre-trained deep learning models. Specifically, InceptionV3 [47] is used to extract image deep features, while AENet [146] is used for audio wave generation as well as feature extraction. SoundNet is another pre-trained audio model leveraging cross modality transfer learning.

Visual Feature Model

To extract visual features from the video keyframes, a popular CNN called InceptionV3 is utilized that is pre-trained on ImageNet (a large scale image dataset). InceptionV3

is an expanded version of original GoogleNet [88] that provides several optimization techniques such as convolution factorization and dimension reduction inside the network. Unlike other well-known deep learning models, Inception expands on both vertical and horizontal directions. In other words, in each Inception, both convolutional and pooling can be employed in parallel. This module not only improves the performance of deep networks but also relatively reduces the computation costs. To handle a large number of outputs in each module, 1×1 conv operations are added before the larger conv layers (e.g., 3×3, 5×5, or 7×7) which act like a dimensionality reduction method. In this paper, first, the last fully connected layer generating the ImageNet 1000 output classes is removed, and then the rest of the network is used as a fixed feature extractor. Specifically, the last average pooling layer in InceptionV3 generating a 2048 dimension matrix is used to compute the final visual features for our dataset.

Audio Feature Models

AENet is a pre-trained model for audio event recognition which is built on a CNN with the capability of operating large temporal inputs. The model is trained on 28 sound classes coming from a wide variety of sources, mostly related to the regular real-world events. That is the best pre-trained model we identified for disaster-related audio feature extraction since most of the public datasets for audio event recognition focus on specific scenarios such as music classification, musical instrument recognition, and speech recognition. The stacked convolutional layers with 3×3 kernels are proposed in AENet to deal with the audio event which might only occur at part of the audio input. The max-pooling layers are indicated as $time \times frequency$. All audios are sampled with the 16kHz sampling rate, 16 bits/sample, mono channel.

SoundNet [147], is another deep learning model targeting sound recognition. It is trained on more than 2 million unlabeled videos by using transfer learning.The deep fea-

tures used as the inputs to our audio model were extracted from the *conv7* layer using their 8 layer model, which show good capabilities to detect high-level concepts, such as natural sounds (water streams, underwater, etc.) and human-related sounds (speech, talking, cheering, etc.). The features form a matrix with size TIME$\times$DIM (5×1024), where TIME is the number of samples in the input audio clip and DIM represents the number of filters that are applied to the *conv7* layer.

Text Feature Model

Global Vectors for Word The word embedding technique in deep learning shows promising performance in NLP. It takes the words or phrases from a vocabulary as input and maps them as vectors into a lower dimensional space. The projection of the words is shared within the whole vocabulary and fine-tuned through back-propagation. Representation (GloVe) [101] are a favorite word embedding technique based on factorizing a matrix of word co-occurrence statistics. The pre-trained model captures the global corpus statistics directly. The word to word vectors represent the meaning of the statistic of word co-occurrence probabilities. In GloVe, it takes the co-occurrence probabilities of two words as a scalar in the similarity matrices and considers the juxtaposition of meanings (e.g., female and male have opposite meanings, but both of them can refer to human beings) as the offsets in the embedding space. For example, consider two words i and j, the relationship of these words can be examined by studying the ratio of their co-occurrence probabilities with various probe words, k. The most correlated words get larger values in the co-occurrence probabilities P. The ratio of P_{ik}/P_{jk} becomes larger if words k related to word i but not word j. The model uses the ratio instead of raw probabilities since a ratio has a better ability to differentiate relevant words from irrelevant words. At the same time, discrimination between two relevant words performs better with the ratio. The weighted least squares objective J is proposed to reduce the difference between the dot product of

the vectors of two words and the logarithm of their number of co-occurrences, where

$$J = \sum_{i,j=1}^{V} f(X_{ij})(\omega_i^T \hat{\omega}_j + b_i + \hat{b}_j - logX_{ij})^2 \tag{4.1}$$

The word vector and bias for words i and j are denoted by ω_i, b_i and $\hat{\omega}_j$, $\hat{b}_j$ respectively. Word j is the word present in the context. X_{ij} is the number of times word i occurs in the context of word j, and f is a weighted function that represents the frequency of the co-occurrences. As co-occurrence counts can be directly encoded in a word-context co-occurrence matrix, GloVe takes such a matrix rather than the entire corpus as input.

4.2 Feature Affinity based Multiple Correspondence Analysis

During a disaster event, the advances and popularity of electronic and mobile devices enable the capturing of a large amount of disaster-related multimedia data [128]. How to effectively and efficiently extract useful information from such disaster-related multimedia data to provide situation awareness information to the general public and the personnel in the Emergency Operations Center (EOC) has become more and more important. Video semantic concept detection, which aims to explore the rich information in videos, uses various machine learning and data mining approaches to address this challenge [5, 29, 35, 95, 96, 116, 148]. In addition, there have been efforts to better bridge the semantic gap between the low-level visual features and the high-level concepts in the literature [149, 150, 151, 152, 153, 154, 155]. Not being restricted to the disaster classification tasks that attempt to classify the disaster scenes from non-disaster scenes, a variety of information relevant to a specified disaster can be utilized, including the hazard situation, recovery progress, disaster effects, and disaster prevention, to name a few. The difficulty increases since all those concepts are surrounding one major premise, which will immensely increase the similarity between the concepts.

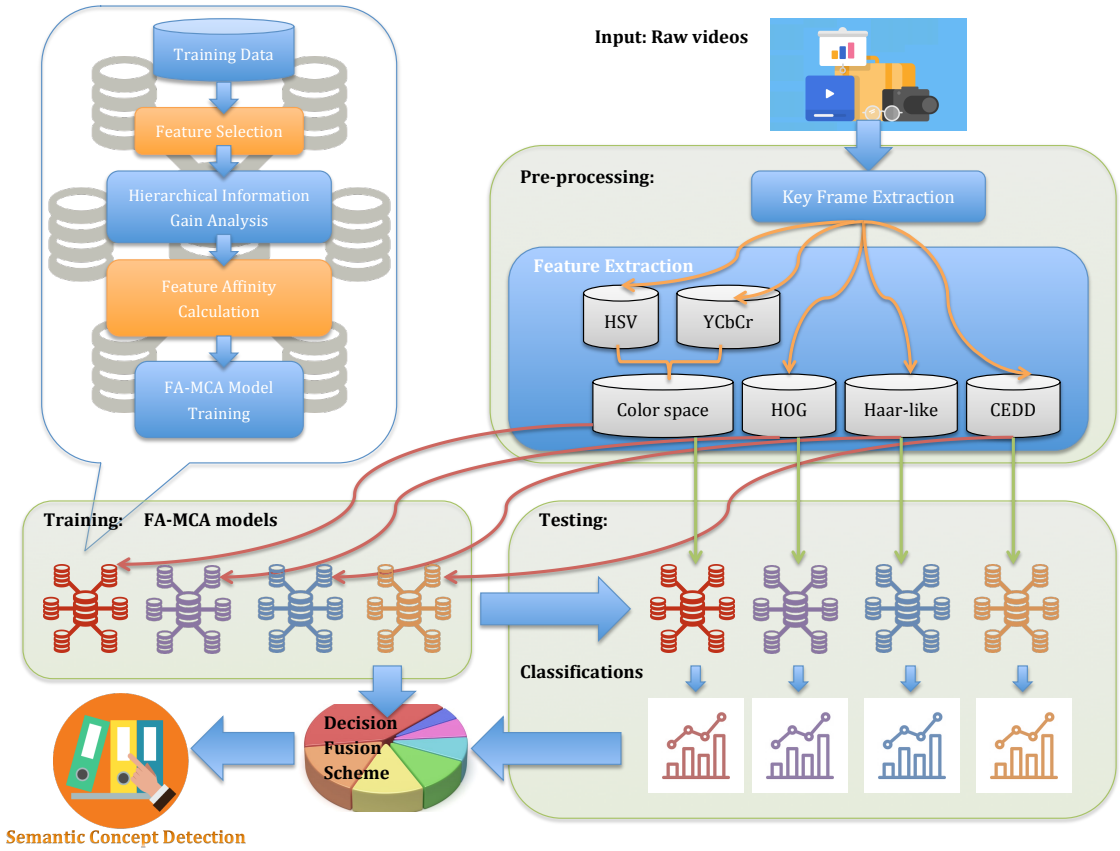


Figure 4.1: Illustration of the FA-MCADF framework

To address such challenges, a Feature Affinity based MCA (FA-MCA) algorithm is first introduced as an individual classifier that outperforms other machine learning algorithms in the disaster-related concept detection tasks. The low-level features are fused into one group after the feature extraction phase and a feature selection method is applied in the FA-MCA model to deal with the high dimensional feature sets. After building a tree-like structure that demonstrates the feature affinities, a weighting function that considers the affinity relationship among the ranks of the features and the number of features at the same rank is developed to improve the MCA algorithm. Furthermore, it is adopted as a basic classifier that can be simultaneously applied to separated feature groups, which reduced the complexity and the computational time. In addition, it has an automatic process to moderate how the weight of a feature dominates the other features.

In the proposed work, FA-MCA is used as the basic classifier and the affinity relationship between the tying features is considered to enhance the classification effectiveness by using conditional weighting functions. It is a scalable framework that accepts a flexible number of feature groups and evaluates the reliabilities of the basic classifiers basing on the evaluation of every learning process. The features are separated into different groups base on the representation levels (e.g., color space, object space, etc.).

The overall framework is illustrated in Figure 4.1. It includes four major steps: pre-processing (the upper right panel), training phase (the middle left panel), testing phase (the lower right panel), and the decision fusion scheme for the final classification. The pre-processing phase includes key frame extraction and feature extraction, which make the data cleaned and structured. In the training phase, the model is trained using the FA-MCA algorithm (details are depicted in the upper right corner) for each structural feature group individually.

The feature affinities are calculated and applied to the final weight as a factor which will be used in the testing phase to classify the testing instances. The proposed framework considers the feature selection procedure as well as the relationship between the features. It will further affect the final weight of each feature and moderate the bias of the classification results. Nevertheless, by distributing the feature set (with an enormously high dimension) into several feature groups (with smaller dimensions) based on the different representation levels (e.g., color space, object space, etc.), a closer dependency analysis on the relationships among the features within each group and between groups can be conducted. For example, from the color space to the object space, the feature groups form a flat structure, indicating that each group is self-structured and relatively independent.

4.2.1 FA-MCA Training Phase

In the training phase, there are two key components: feature selection and feature affinity calculation. The FA-MCA model includes the chi-squared test [156] to evaluate and select the most representable feature values if the dataset is in a very high dimension. By building up a decision tree structure that uses the reduced features, the useful information and positions are stored and utilized in the feature affinity calculation component.

The proposed feature affinity calculation component assigns the weight of each feature based on the position of the feature ($depth_i$) in the tree structure. Furthermore, the number of features at the same depth in the tree is also considered as useful information. By considering the number of features at the same depth, the weight assigning to each feature at the same rank will be reduced. It is obvious that the feature, which holds the rank by itself, should be more valuable than those features at the same depth. Unlike the information gain in the original decision tree algorithm, the relationships between features in the structure are preserved after the feature selection component to make the final MCA weight generation multivariate.

As a result of feature selection, the total number of useful features in the training phase decreases. The number is also considered while calculating the feature affinity (FA_i) for feature index (i). However, instead of considering only the ratio of feature reduction, the proposed feature affinity calculation component utilizes the position of the feature to eliminate the effect. (as shown in Equation (4.2)). It will be directly applied to the feature that is responsible for the decision in a certain level only by itself. In other words, there is no other feature competing with the current one while making the decision. Let I_{orig} and I be the total number of features before and after feature selection, respectively. The

natural logarithm is used to obtain a simpler derivative under the curve $y = 1/x$.

$$FA_i = \frac{1}{log_e(depth_i + 1)} + \frac{I_{orig}}{I}. \quad (4.2)$$

$$Share\_FA_i = \frac{FA_i}{\#\ of features\ in\ depth_i}. \quad (4.3)$$

For each selected feature, the feature index (i) and the feature level ($depth_i$) are recorded. They are reused here for the feature weight calculation. The number of features holding the same rank will be counted to evaluate how those features in that rank dominate the other features. In brief, dividing the count will decrease the respective affinity. Such modification is shown in Equation (4.3).

The feature affinity is supposed to improve the final classification results due to the deep observation of the correlations between features. That is, the relationship between the features plays an important role to make the feature domination consistency. Without such information, each feature that is considered as independent will enlarge the weighting effect. The most direct influence is that more instances will be classified to be either positive or negative during the testing phase since some features are over-weighted.

By integrating the feature affinity with the MCA algorithm, the final weighting function of the MCA algorithm is thus modified. For the details of how to generate the original MCA weighting matrix, please refer to [40]. After selecting each feature to calculate the MCA weight, a 3-D matrix (MW) is generated as a form of feature-value pairs. For each pair of the feature and class, the final weight is multiplied by the feature affinity. The function is shown in Equation (4.4).

$$MW_i^c, \varphi = MW_i^c, \varphi * Share\_FA_i, \quad (4.4)$$

where c represents the class of the instance, and φ represents the feature value. Similar to Equation (4.2), i' indicates the feature index after feature selection.

4.2.2 Testing Phase

The final weighting matrix generated during the training phase is used in the testing phase in order to get the final ranking scores for the testing instances. Those ranking scores are responsible for predicting the concept class. The ranking procedure starts with adding all feature weights for instance t, and calculates its average [40].

For classification, all the ranking scores of the testing instances are sorted in the descending order, and the top instances are selected with the best selection threshold [157].

Since the testing phase sums up the feature weights learned from the training phase, the proposed feature affinity calculation will make the final weight of each feature more durable and improve the testing results.

4.2.3 Experimental Analysis

Dataset Description

Although the MCA-based framework can be used as a general framework that works for various multimedia application domains, in this work, the specific task of detecting disaster-related semantic concepts is selected using a dataset obtained from the Federal Emergency Management Agency (FEMA) website[1]. Since the semantic concepts obtained from this website are different from the normal disaster event concepts, it is more useful to examine the effectiveness of the proposed framework.

The dataset contains over 200 videos and thousands of key frames that are related to seven different concepts. However, there are still many similarities between some of the concepts. The statistics information is shown in Table 4.1 which depicts the name, number of positive instances, and number of videos of each concept. When the similarity between concepts increases, the task of concept detection becomes more challenging.

[1] https://www.fema.gov

Meanwhile, the weight generation of each feature needs a higher accuracy to improve the training and testing performance. These are the reason and motivation for proposing the FA-MCA approach.

Hence, the dataset consists of data instances at the frame level with the binary class information. The finalized dataset is then split into training and testing sets using three-fold cross-validation [158] based on the count of videos. In other words, the entire data set is divided into 3 different folds with approximately 1/3 of the videos (one fold) for testing and 2/3 of the videos (two folds) for training purpose. Specifically, for video concept detection, a set of key frame instances that belong to the same video is assigned to either the training dataset or testing dataset during the separation in order to preserve the information between frames that represents in each video.

Table 4.1: FEMA dataset statistics: number of key frames in each concept

No.	**Concepts**	**Positive Instances**
1	Flood	258
2	Human Relief	92
3	Damage	281
4	Training Program	148
5	Disaster Recovery	369
6	Speak	1230
7	Interview	117
	Total	2495

4.2.4 Evaluation Results

The performance evaluation takes the precision, recall, and F1-score values as the criteria [159]. Table 4.2 presents the experimental results in details, while the proposed FA-MCA algorithm shows the best performance on average in comparison with ANNs, SVMs, DTs, and LR classifiers (available in WEKA [160]). All the classifiers are tuned to achieve their best performance during the experiment, and the results are ordered by

the average F1-scores (the last column in Table 4.2). SVMs and ANNs are two examples of black-box models that can only be verified externally. They are always popularly used in different domains where good classification performance is preferred. However, from the evaluation results, it can be interpreted that their discriminating power is not significantly better than the other models, which means for this specific dataset, a more accurate model is needed to differentiate the concepts. The Radial Basis Function (RBF) network is selected as a representative of ANNs since it performs better than other ANNs classifiers on this specific dataset. As can be inferred from this table, the improvement of the average F1-score of FA-MCA is around 10% when comparing to LR, which achieves promising results in comparison with the other classifiers. LR is a statistical method that is always compared to the ANNs models in many classification tasks. It shows its capability of handling a dataset with a small number of positive instances (i.e., imbalanced data). On the contrary, the RBF Network reaches 86.07% for concept "Speak", which is the most balanced concept, but it is still 1% worse than FA-MAC. Compared with the other machine learning methods mentioned here, DTs take the information gain values as the common criterion and have the advantage that each tree can easily be expressed as rules. FA-MAC also takes the information gain values as one of the feature selection criterion and avoids the disadvantage of DTs, which is losing information along the splitting process. In addition, The FA-MAC algorithm shows significant improvements (12% and 13%, respectively) on the complicated semantic concepts such as "Human Relief" and "Training Program".

4.3 Multiple Correspondence Analysis based Neural Network

In the Internet age, the volume of multimedia data (including video, audio, image, and text) grows exponentially, carrying a variety of valuable information [154, 161, 162].

Table 4.2: FA-MCA algorithm's performance on the FEMA dataset

		Flood	Human Relief	Damage	Training Program	Diasater Recovery	Speak	Interview	Average
RBF Network	**Pre**	70.07%	1.10%	71.77%	35.20%	36.97%	78.23%	35.50%	46.98%
	Rec	51.30%	33.33%	78.60%	34.87%	47.60%	99.93%	40.27%	55.13%
	F1	36.83%	2.17%	62.47%	6.47%	26.53%	86.07%	15.53%	33.72%
SVM	**Pre**	70.17%	1.67%	71.77%	1.87%	70.33%	82.93%	68.83%	52.51%
	Rec	46.20%	32.33%	65.97%	33.33%	66.87%	88.57%	40.70%	53.42%
	F1	29.47%	3.17%	52.57%	3.53%	50.87%	82.27%	16.83%	34.10%
Decision Tree	**Pre**	70.13%	1.43%	72.10%	68.60%	70.37%	82.93%	68.77%	62.05%
	Rec	45.23%	32.33%	61.40%	40.87%	61.67%	81.37%	40.53%	51.91%
	F1	29.77%	2.73%	49.13%	17.20%	46.47%	77.67%	18.90%	34.55%
Logistic Regression	**Pre**	70.23%	67.87%	71.77%	68.60%	71.17%	82.97%	68.83%	71.63%
	Rec	58.23%	38.07%	63.73%	49.23%	58.17%	81.00%	44.33%	56.11%
	F1	42.97%	11.13%	50.60%	32.97%	47.93%	77.47%	22.80%	40.84%
FA-MCA	**Pre**	70.25%	34.18%	71.91%	68.61%	70.32%	82.91%	68.79%	66.71%
	Rec	61.67%	30.52%	72.92%	65.48%	68.90%	97.78%	48.95%	63.74%
	F1	46.62%	24.45%	64.80%	45.07%	52.69%	87.42%	29.54%	50.08%

Multimedia data can be accessed from different kinds of devices, making it more convenient for people to get a visual understanding of the situations that they care about [125, 163]. Video semantic concept detection, which aims to explore the rich information in the videos, uses various machine learning and data mining approaches to address this challenge [135, 164, 165]. In addition, many existing approaches are making every effort to better fill in the gap between the low-level visual features and the high-level concepts [149, 166, 167].

In this section, a novel framework of Multiple Correspondence Analysis based Neural Network (MCA-NN) is proposed to address the challenges in shallow learning. It integrates the Feature Affinity based Multiple Correspondence Analysis (FA-MCA) models into one large neural network model. The major contributions of this work are as follows:

- First, this is the first time that the MCA-based model is applied to separated groups of features and generates higher-level features as the inputs of the deep learning component;
- Second, the proposed semantic concept detection framework is utilized to decide the video concept instead of frame-based classification;

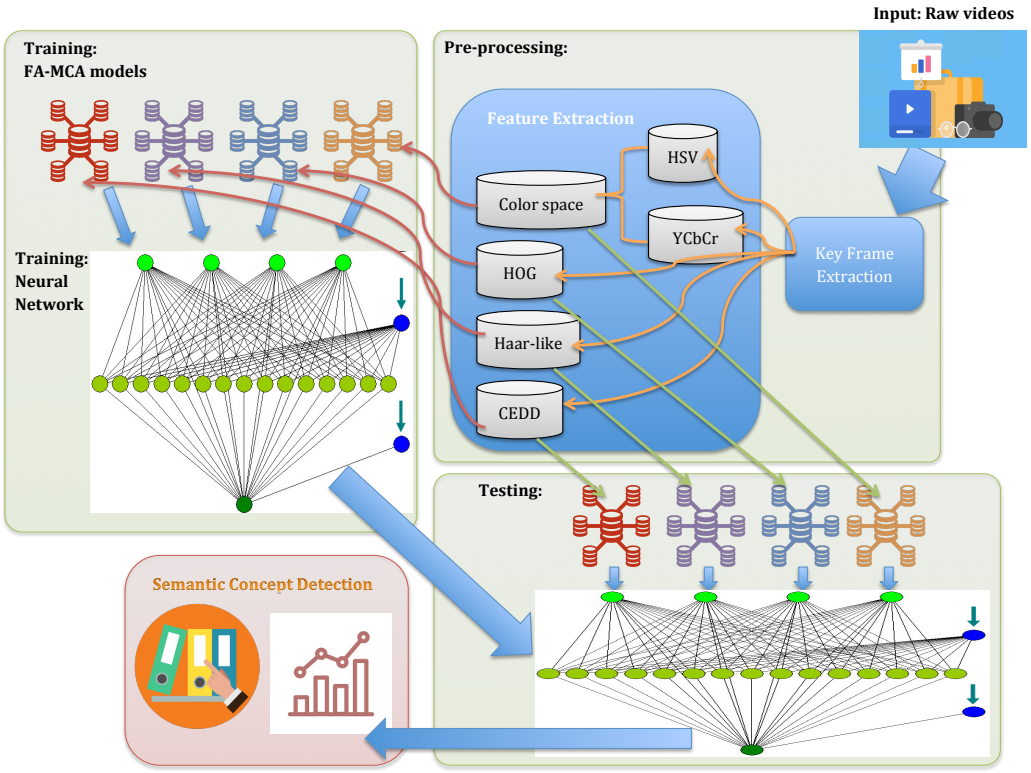


Figure 4.2: Illustration of the MCA-NN framework

- Furthermore, the process of deciding the neural network module is automatic. The most important parameters building the network are obtained from the outputs of the FA-MCA models and the corresponding statistical information.

In the MCA-NN framework, the input representations of a low-level feature are transformed into a higher-level value using FA-MCA model training. However, it is one stage feature transformation, which is considered as shallow learning while high-level features are more global and more invariant. To address this issue, it is worth considering the MLP neural network, which takes the transforming features to the predictor.

The overall framework is illustrated in Figure 4.2. It includes three major steps: pre-processing (the upper right panel), training phase (the upper left panel), testing phase (the lower right panel). The output classification results from the network are frame based.

The final classification of the framework concludes the single frame decisions for each video to produce the entire video classification.

The pre-processing phase includes key frame extraction and feature extraction, which make the data cleaned and structured. In the training phase, the model is trained using the FA-MCA algorithm for each feature group independently. The low-level features were learnt through each FA-MCA model and transformed into a higher-level feature. Each model produces one ranking score for each instance, and the ranking score is normalized as a new feature that includes a higher level semantic. Followed by the FA-MCA model training, an MLP network is created using the FA-MCA outputs to deeply learn the relationships between high-level features.

The low-level feature value affinities are calculated and accumulated as weighting factors, which will be used in the testing phase to generate the high-level feature value of the testing instances. The low-level feature sets are distributed into different groups for high-level feature value extraction based on the different representation levels (e.g., color space, object space, etc.). For example, from the color space to the object space, the feature groups form a flat structure, indicating that each group is self-structured and relatively independent. Afterward, the outputs from the FA-MCA models are utilized as inputs of hierarchical feature learning network, which makes use of the relationships between independent high-level feature values.

4.3.1 MCA-NN Training Phase

In the training phase, there are two key components: low-level feature transformation using FA-MCA and the MLP neural network that takes the transformed feature values as inputs; and feature transformation component uses FA-MCA to produce a single value for each feature group, which leverage the low-level feature group into a more abstract rep-

resentative value. The calculation bases on a weighting matrix that takes each low-level feature into consideration during learning process. The neural network builds up based on the outputs and the statistics of the training results from each FA-MCA model. The output values are ranking scores for training instances that can be used for classification purpose. However, in this proposed work, the ranking scores are used as higher-level features for the following deep learning process.

To fully utilize the value in each output, the number of hidden layers is decided by the number of input layers. Considering the permutation of N optional input layers, the full permutation of the selection is $N!$. One bias weight (w_{i0}) is included in the total number of hidden neurons and will not be updated during the back propagation, as well as the one counts for the input neurons. For example, in the proposed framework, there are four input layers for different low-level features. As taking variable N as 4, there are 5 input weights in total, which are 4 weights for different inputs plus one bias weight. For the hidden layers, there are $4! = 24$ weights plus one bias weight.

The $tanh$ activation function is used to enable a wider range of output instead of linear activation, the input neurons (a_i) for next layers are calculated base on the following formula:

$$a_i = tanh(net_i) = \frac{e^{net_i} - e^{-net_i}}{e^{net_i} + e^{-net_i}} \tag{4.5}$$

The $tanh$ function restricts the output between -1 and 1, which can be used to predict the event if the value turns out to be positive or negative. Therefore, the transformed features are normalized between -1 and 1 as well. If the transformed feature is closer to -1, it means the FA-MCA model has learnt that the low-level features for the specific instance are more likely to represent the target concept, vice versa.

In formula 4.5, net_i is the correspondence neuron output of current layer, which accumulates the weighted output from the previous layers (shows in formula 4.6), $Pred(i)$ is the set of all neurons j for a connection $j \rightarrow i$ exists, called the set of predecessors.

$$net_i = w_{i0} + \sum_{j \in Pred(i)} w_{ij} a_j \tag{4.6}$$

The initial weights for the calculations of all the hidden layers use the F1 scores from the FA-MCA training results, which are the values between 0 and 1. In that case, the initial weight will be large if the transformation shows high confidence by a large F1 score. A smaller weight will be assigned if the confidence of specific FA-MCA training model is lower. Therefore, the transformed high-level features might not be able to carry out a well learnt concept comparing with other features. To obtain better initial weights for each input, the best F1 scores for the training dataset using the FA-MCA models are modified to fit in the range of [-0.5,0.5]. In order to get an initial output between [-1,1], each weight is divided by 2 (E.g., 4 input layers with each weight between [-0.25, 0.25]). The bias weight takes the average F1 score for all low-level feature transformation models and modified it also fit in the same range of initial weight.

For the output layer (Dark green neuron in figure 4.2), all the weights for calculating the hidden neurons are initiated randomly following the requirement of range in [-0.5, 0.5]. Since each training data set is unique for each concept, select the weights randomly will not restrict too much to the output. However, it needs several rounds of back propagation to compute gradients. The repeating process is set to 10,000 times during experiments for regular runs to have error plummets. The error rate is accumulated by p training instances and calculated based on formula 4.7 once of the training cycle to determine the learning rate of the output layers for the next cycle, which is used in the process of back propagation.

$$E_{total} = \frac{1}{2}\sum_{i=1}^{p}(target_i - output_i)^2 \tag{4.7}$$

The weights are updated during the back propagation in order to have the actual output ($output_i$) to be closer to the target output ($target_i$). Namely, minimizing the error for each hidden neuron and the whole network. The changes of weights (w_i) in the output layer calculation affect the total error by taking the partial derivation as following:

$$\frac{\partial E_{total}}{\partial w_i} = \frac{\partial E_{total}}{\partial output_i} * \frac{\partial output_i}{\partial net_i} * \frac{\partial net_{ti}}{\partial w_i} \tag{4.8}$$

The partial derivative of the activation function is 1 minus the square of the current layer output (shows in formula 4.9).

$$\frac{\partial output_i}{\partial net_i} = 1 - tanh^2(net_i) \tag{4.9}$$

The backward calculation of the weight changes for hidden layers is similar but slightly different to account the output of each hidden layer neuron contributes to the output neuron. So every hidden layer weight change is the partial derivative of the total hidden layer input with respect to each weight (w_{ji}), where j is the total number of input neurons:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_{ji}} =&(\sum_{j} \frac{\partial E_{total}}{\partial output_j} * \frac{\partial output_j}{\partial net_j} * \frac{\partial net_j}{\partial output_{ji}}) \\ &* \frac{\partial output_{ji}}{\partial net_{ji}} * \frac{\partial net_{ji}}{\partial w_{ji}}\end{aligned} \tag{4.10}$$

Both hidden layers' and output layers' weights are updated during the runs to decrease the error by multiplying by a learning rate, the following formula shows the update step, where w_i^+ represents the updated weight:

$$w_i^+ = w_i - \eta * \frac{\partial E_{total}}{\partial w_i} \tag{4.11}$$

The learning rates η for both updating functions (output layer and hidden layers) are set to 0.7 empirically at the initial step. However, in some of the training process, 0.7

seems too large to tighten up the errors. It takes so many learning cycles but still could not be able to find a proper prediction value with a low error rate. The proposed framework automatically detects the large error rates after the first 1000 runs as tolerance. If the total error remains greater than 0.01, the learning rate of the output layer will be reduced by 10 times (reset to 0.07). Consequently, since the learning rate affects the duration of the learning process, the training cycle extended two times longer than the original one to acquire an output prediction value with an acceptable error rate.

4.3.2 Testing Phase

The final weighting matrix generated during the training phase of FA-MCA is used in the testing phase in order to get the final ranking scores for the testing instances. Those ranking scores are responsible for representing the high-level concepts. The ranking procedure starts with adding all feature weights for instance t, and calculates its average value [40].

For the purpose of feeding the variables into the well-trained neural network, all the ranking scores of the testing instances are normalized between [-1,1] as the training instances in order to better represent the value that is similar to the output of the $tanh$ function.

Since the best F1 score for the training data can be calculated by attempting to separate the transformed low-level features into the positive class (containing the target concept) or the negative class (not containing the target concept), the F1 score for each FA-MCA model is recorded as the confidential variable that can be utilized for initializing the MLP weights.

The well-trained MLP network is directly used by feeding all the testing instances one by one to generate the prediction values. As all the weights are updated and fixed during the training phase to optimally derive the positive instances from the negative

instances, the testing phase is as easy as running the fixed network to compute the output. Same to the ideal distribution in the training phase, a smaller output value in the range of [-1,1] predicts a positive instance, while a larger output value predicts a negative one. The number 0 is selected as the value to do the classification, which means the instance holding a prediction value smaller than zero will be classified as positive.

4.3.3 Semantic Concept Detection

As mentioned earlier, the final semantic concept predictions are concluded by the count of the videos. In that case, the output from the neural network, which has the classification results for each individual frame, needs to be integrated to get the finalized decision of each video. The framework takes the classification results of the frames for one video to decide the final classification. By counting the total number of frames for one video that are being tested, the portion of the predicted class (negative or positive) affects the final decision. During the experiments, the portion threshold is set to be 0.6, which means if there are more than 60% of the frames being classified as negative, the video will be classified as negative. Otherwise, the video will be predicted as positive. The negative labeled video means that the target semantic concept is not detected from the tested video. On the contrary, if the video is classified as positive, it means that the concept is detected. The experiments of how to decide the threshold is shown in the following section.

4.3.4 Experimental Analysis

Dataset Description

In this work, a specific task of detecting disaster-related semantic concepts is selected using a dataset obtained from the Federal Emergency Management Agency (FEMA) website, although the framework can be used as a general framework that works for various

multimedia application domain. The semantic concepts obtained from this website are different from the normal disaster event concepts. It is more useful to examine the effectiveness of the proposed MCA-NN framework that improves the capability of detecting the differences between similar concepts.

The dataset includes more than 200 videos, which contain thousands of key frames that are related to seven different concepts. However, there are still a great amount of similarities between the concepts. The statistics information is shown in Table 4.3 that depicts the name, the number of positive instances, and the number of videos of each concept. When the similarity between concepts increases, the task of concept detection becomes more challenging. Meanwhile, a well trained neural network for the transformation of features improves the training and testing performance. These are the reasons and motivation of proposing the MCA-NN framework.

Table 4.3: FEMA dataset statistics: number of videos in each concept

No.	**Concepts**	**Positive Instances**	**Videos**
1	Flood	258	21
2	Human Relief	92	4
3	Damage	281	21
4	Training Program	148	7
5	Disaster Recovery	369	16
6	Speak	1230	145
7	Interview	117	23
	Total	2495	237

Figure 4.3 also depicts the samples of each concept in details on which are the key frames extracted from the videos and used during evaluation process. It is easier to differentiate the concept “Flood” in Figure 4.3a from the concept “Human Relief” in Figure 4.3b than to distinguish the concept “Speak” in Figure 4.3f from the concept “Interview” in Figure 4.3g.

(a) Flood (b) Human Relief (c) Damage (d) Training Program

(e) Disaster Recovery (f) Speak (g) Interview

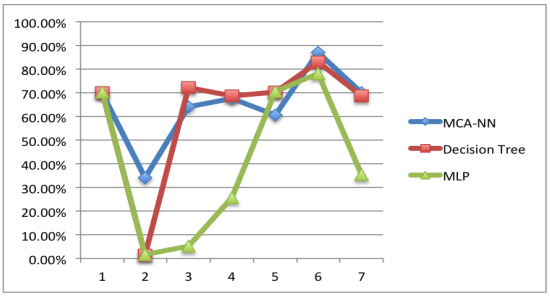
Figure 4.3: Sample images represent seven different concepts in the FEMA dataset

Evaluation Results

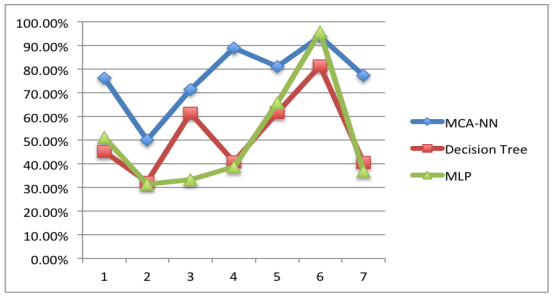
The performance evaluation takes the precision, recall, and F1-score values as the criteria [159], which consider the number of positive and negative instances in each class. The F1-score measure is considered as the most valuable comparison metric since it is the trade-offs between the precision and recall values. All the classifiers are tuned to achieve their best performance during the experiment.

The proposed framework shows the best performance on average in comparison with the decision tree and MLP classifiers (available in WEKA [160]). The performance by each comparison criterion is illustrated in Figure 4.4. Each plot takes the concept id as the x-axis and the percentage evaluation result as the y-axis. The concept id that refers to a different concept name can be found in Table 4.3. It is clear that, during the comparison of each criterion, the proposed method wins most of them, especially in the comparison of the recall and F1-score values, which are in Figure 4.4b and Figure 4.4c, respectively.

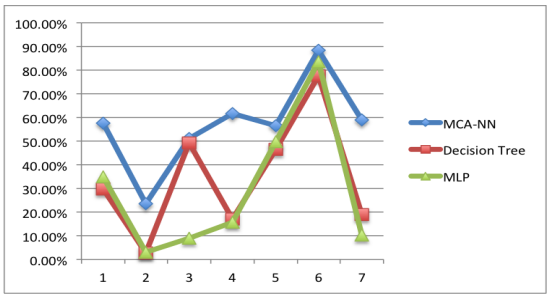
Table 4.4 presents the experimental results in details. As can be seen from this table, the improvement of the average F1-score is more than 27% when comparing to MLP.



(a) Precision



(b) Recall



(c) F1

Figure 4.4: MCA-NN model performance by each evaluation criteria: precision, recall, and F1 score

Table 4.4: Comparison results of the MCA-NN algorithm's performance on the FEMA dataset

	Decision Tree			**MLP**			**MCA-NN**		
Concepts	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Flood	70.13%	45.23%	29.77%	**70.27%**	51.03%	34.90%	69.74%	**76.19%**	**57.64%**
Human Relief	1.43%	32.33%	2.73%	1.63%	31.30%	3.07%	**33.99%**	**50.00%**	**23.51%**
Damage	**72.10%**	61.40%	49.13%	5.10%	33.33%	8.87%	64.14%	**71.43%**	**51.20%**
Training Program	**68.60%**	40.87%	17.20%	25.67%	38.56%	15.61%	67.51%	**88.89%**	**61.65%**
Disaster Recovery	70.37%	61.67%	46.47%	**70.57%**	65.83%	49.87%	60.53%	**81.11%**	**56.64%**
Speak	82.93	81.37%	77.67%	78.23%	**95.57%**	83.67%	**86.92%**	93.88%	**88.49%**
Interview	68.77%	40.53%	18.90%	35.53%	36.77%	10.30%	**70.09%**	**77.38%**	**59.01%**
AVERAGE	62.05%	51.91%	34.55%	41.00%	50.34%	29.47%	**64.70%**	**76.98%**	**56.88%**

Compared to the Decision Tree, the average results (precision, recall, F1 score) improve 2.65%, 25.07% and 22.33%, respectively. Although the MLP recall performs nearly two percent better than MCA-NN for one of the concepts (i.e., Speak), it does not get the best F1 score, which means it takes as many instances as positive; while more negative instances are wrongly classified. Also, it shows poor performance when the number of positive instances is very small (i.e., imbalanced data). However, MCA-NN performs well, no matter whether the number of positive instances is large or small in a dataset.

Additionally, since we prefer to recognize as many related events as possible for the purpose of disaster information analysis, the recall values earn more attention when comparing to the precision values. However, blindly increasing the number of positive instances in the classification process could only bring a higher recall value. A better F1 score relies on a more accurate classification framework. In other words, the increasing recall values at the cost of the precision values would not be able to get a stable F1 score in the experiments.

Figure 4.5 shows the experiments on selecting the best threshold of making decisions for entire video classifications. It is clear that the precisions are affected slightly during the test. The rightmost three bars, which represents taking 0.6 as the threshold that is used for all the experimental results depicted in Table 4.4, show the best recall and F1 score values in this test. From the test, we can conclude that since the precision values would not

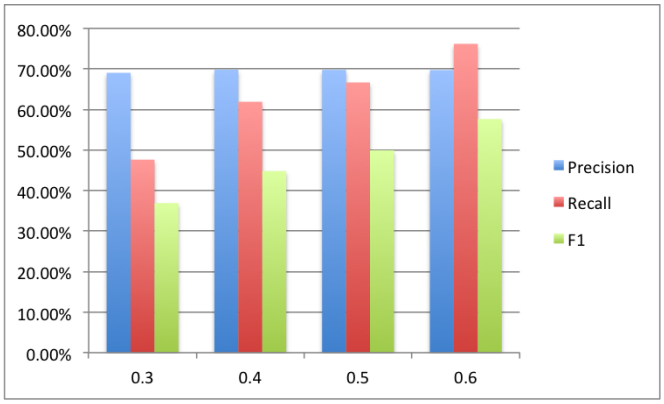


Figure 4.5: The experimental results for deciding the video classification threshold

be greatly affected by the threshold, it would be better to increase the threshold in order to get the best recall and F1-score values. The bar chart shows a gradually increasing trend for both recall and F1-score values, accompanying with an increasing threshold. However, when the threshold comes to 0.7, the precision value suddenly dropped to 0 in the test. So the final threshold is determined to be 0.6.

4.3.5 Conclusions

Disaster-related concept detection includes disaster event detection, disaster preparation training, disaster recovery, and disaster damage situation, to name a few. Since it does not limit to the straight forward disaster events, the concepts that need to be utilized are varied for the aim of managing the disaster information. Since the correlations between those concepts are higher than the diverse disaster events, it makes the classification task more challenging. To tackle this challenge, in this paper, the MCA-NN framework is proposed to convey the low-level features into the higher-level feature values through the FA-MCA models, considering the relationship between the features within each feature group. The shallow network learned and transformed features were used as the input for a deeper learning neural network for further training purpose. As a result, critical low-

level features are memorized and depicted as the higher-level features. Consequently, the higher-level features are explored in details to better understand the concepts.

Comparing with the decision tree and MLP classifiers, the experimental results show significant improvements for all the evaluation criteria, which means that the proposed framework successfully transformed the low-level features and truly learnt the concepts when differentiating the interrelated concepts. However, there is still some improvements that can be further carried out.

In the future, this framework will be further extended and tested for more concept detection applications. It is worth considering to do more research on the randomly assigned initial weights in order to reduce the repeating cycles. Other neural networks and back propagation algorithms can be utilized to better fulfill the deep learning purpose.

4.4 Automatic Convolutional Neural Network Selection for Image Classification Using Genetic Algorithms

The power of transfer learning in visual data analytics has been extensively pointed out in the literature [49, 168, 169]. Existing deep learning models have millions of parameters which require immense computing power and very large-scale datasets to be trained from scratch [53, 88]. Transfer learning is the solution to mitigate this problem by utilizing part of the pre-trained models as the starting point of training a related task. By using transfer learning and powerful image classifiers trained on huge datasets (e.g., ImageNet), it is possible to effectively train a deep learning model on regular datasets with thousands rather than millions of training samples. However, the questions are (1) How can one determine the most efficient pre-trained model for each dataset? and (2) Will the size of the dataset or the nature of the data affect the classification results? For example, in our previous work on video event detection [49], AlexNet [87] performs better than

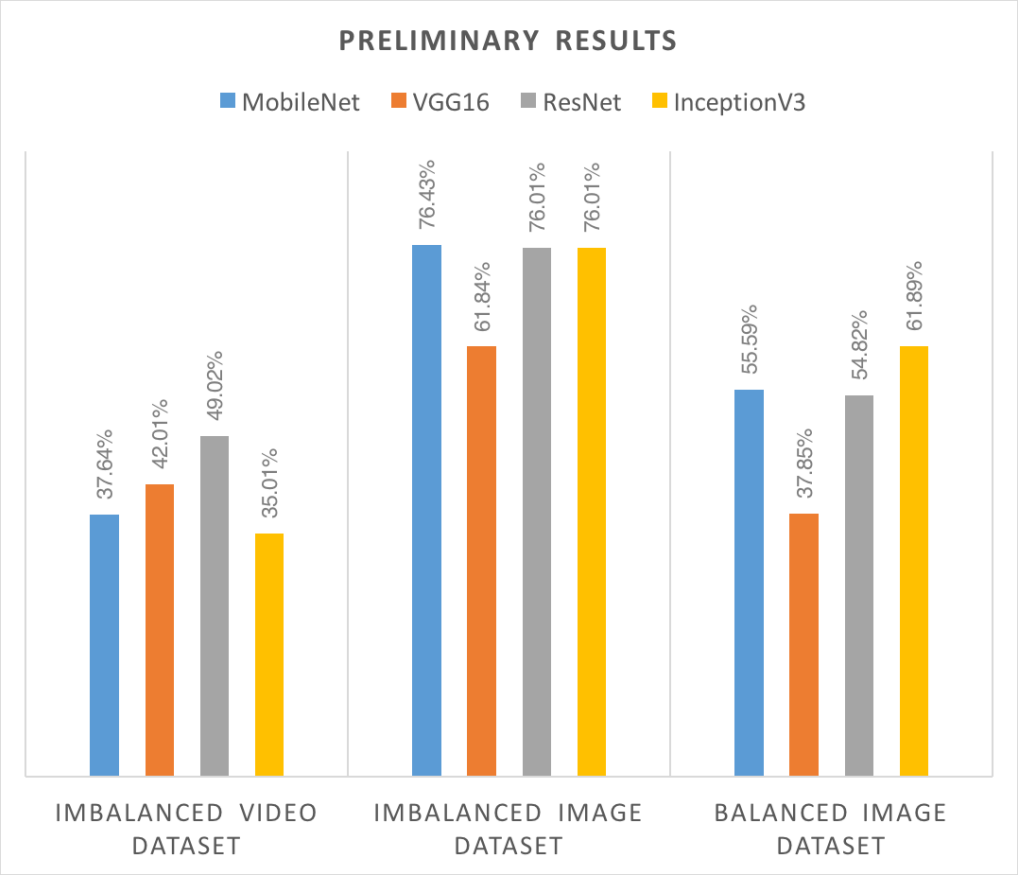


Figure 4.6: The accuracy of four pre-trained deep learning models on three different datasets.

the advanced models such as GoogleNet [88] on the disaster-related video dataset, while ResNet achieved the best performance in a public dataset called TRECVID SIN [170]. In addition, based on our preliminary results (as shown in Figure 4.6), Inception-v3 [47] performs well for a balanced dataset like CIFAR10 but ResNet or MobileNet [54] may perform better on imbalanced data. This inconsistency is mainly due to the level of similarity between the source (e.g., ImageNet) and target (e.g., disaster or TRECVID) datasets. Other factors such as the distribution of data, size of the dataset, and resolution of images may also affect the performance of each model. Thus, in this study, we try to answer these questions using an optimization algorithm.

Deep neural networks such as Convolutional Neural Networks (CNNs) have achieved several significant milestones in visual data analytics. Benefited from transfer learning, many researchers use pre-trained CNN models to accelerate the training process. However, there is still uncertainty about the deep learning models, structures, and applications. For instance, the diversity of the datasets may affect the performance of each pre-trained model.

Genetic algorithm (GA) is a subset of Evolutionary Algorithms (EA). It is ordinarily used for search and optimization problems using biogenetic operations such as selection, mutation, and crossover [171]. In recent years, integrating GA with deep learning has been attracting significant attention. More specifically, it is utilized for automatic selection of hyper-parameters (e.g., learning rate), parameters (e.g., kernel size), and network structures [69, 72].

Different from existing work, we utilize GA to enhance the performance of CNNs in image classification tasks using transfer learning in this work. Specifically, several existing pre-trained models are selected as the original population and then GA is utilized to automatically select the best model from the population or to regenerate the new candidates using GA operations. To serve this purpose, a new encoding model representing the pre-trained models in the population is presented. The GA model selects the best CNN model and extracts the corresponding features from that model. The fitness function employed in the GA algorithm is F1-score which is regularly used for the evaluation of imbalanced datasets.

Figure 4.7 depicts the overall structure of the proposed framework which includes the genetic code generation and revolution process. In each generation, a set of pre-trained CNN models are selected. The items included in a particular set is determined by the individual's genetic code. Consequently, one individual represents a possible combination of CNN models that will be utilized to generate the deep features. Then, a linear SVM

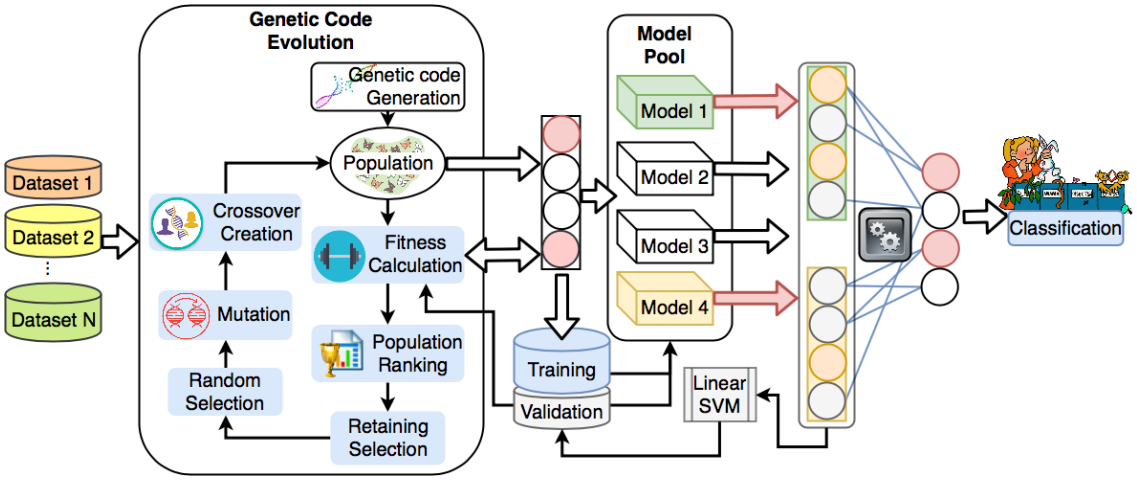


Figure 4.7: A genetic algorithm-based framework for deep neural network selection

classifier is trained on the simply concatenated deep features to validate the effectiveness of the model selection using a fitness function. The fitness function defined in the genetic model takes the average F1-score from the validation data as the feedback to evaluate the individual's rank in the current population. After several iterations of genetic operations, the best individual with the highest fitness score will be selected as the optimal solution for our problem. Then, an automatically created network with several dense layers is employed to leverage the deep feature representations and generate the final classification results.

A step-by-step explanation of the proposed framework is described as follows.

4.4.1 Genetic Code Revolution

Our network selection problem can be compared to a black box switching problem. Suppose, the black box device contains a bank of four input switches which refer to the four selected deep learning models. The output can be represented as $F = f(s)$, where s is a particular setting of the four switches and $f(\cdot)$ is the fitness function. The objective of the problem is to set the switches to obtain the maximum possible F value. In our model, we

are aiming to get the maximum Average (Avg.) F1 measure, thus, it is used as the fitness function. Considering a group of four binary numbers, a total of 16 (2^4) different combinations from "0000" to "1111" can be generated. The combination "0000" means no model is going to be selected since there is no output signal. By removing this individual from the gene pool, there will be in total 15 different genes that lead to various output performances. Exhaustive search or brute-force search solutions are very time consuming for this problem and the complexity increases exponentially when one more model is added to the model pool. Therefore, utilizing GA is a smart method that improves the process of approaching the optimal solution in a shortcut. The entire evolution algorithm is illustrated in algorithm 1.

One iteration of the genetic code revolution consists of several genetic operations as explained below.

- **Initialization:** With GA, we first code the selected model set (switches) as a finite-length string. A simple code can be generated by considering a string of four 1's and 0's where each of the four switches is represented as "1" if the model is selected and "0" if the model is discarded. For instance, with this schema, the string "1001" encodes the setting where the first and the last switches are on while the others are off. First, the initial population is randomly selected. Then, we define a set of genetic operations that takes this initial population and generates successive populations. The new populations can be potentially improved over the time.
- **Fitness Calculation:** As we mentioned before, the function f can be considered as a measure of profit, utility, or goodness that we want to maximize. Copying genetic code according to the fitness values means a code with a higher value has a higher probability to contribute to the next generation's offspring. This operator is an artificial version of natural selection, Darwinian survival theory. For each generation, we take a portion of the best performing individuals as judged by our fitness func-

Algorithm 1: Genetic code evolution

$RETAIN \leftarrow 0.2$
$RANDOM\_SELECT \leftarrow 0.1$
$MUTATE \leftarrow 0.2$
for *individual* $i \in$ *Population* p **do**
    calculate Fitness function $f(i)$
    $grade[i] \leftarrow f(i)$
Sort $grade$ in descending order
for $x \in [0, RETAIN * len(grade) - 1]$ **do**
    $parents.append(grade[x])$
# Random selection
for $x \in [RETAIN * len(grade), len(grade) - 1]$ **do**
    if $RANDOM\_SELECT > random()$ **then**
        $parents.append(grade[x])$
# Mutation
for $pa \in parents$ **do**
    if $MUTATE > random()$ **then**
        $POS \leftarrow Randint(0, len(pa) - 1)$
        Flip code pa at positon POS
Crossover
$size \leftarrow len(Population) - len(parents)$
while *len(children)* $<$ *size* **do**
    select $famale$ and $male$ randomly from $parents$
    if $female \neq male$ **then**
        $child =$
          $(male[0, len(male)/2 - 1] + female[len(male)/2, len(male) - 1])$
        $children.append(child)$
$parents.append(children)$
return $parents$

tion. These high-performers will be the parents of the next generation. Instead of running through the whole framework and getting the feedback from the validation set, we proposed to train a linear SVM classifier before building the deep neural networks as a shortcut to validate each individual in the current population. In this case, we can significantly reduce the computational complexity. Based on our preliminary results, Linear SVM classifier outperforms all the other simple classifiers, such as decision tree, RandomForest, etc. Therefore, it is utilized to calculate the fitness score and potentially ensures the reliability of the final output. Since we aim to tackle the multi-class classification task in this paper, the evaluation metrics and SVM classifier are evaluated based on the one-vs-rest decision function, which is defined in *scikit-learn* [172]. The fitness function $f(s)$ for each dataset that includes C classes is calculated in Equation 4.12.

$$f(s) = (\sum_{c=1}^{C} \frac{2 * truePostive_c}{2 * truePositive_c + False_c})/C, \tag{4.12}$$

where $truePostive_c$ and $False_c$ represents the number of instances that are correctly predicted as concept c and the total number of wrongly classified instances, respectively.

- **Grades Ranking and Population Retaining:** Each individual in the current generation will get a rank based on the descending order of the fitness scores. The retaining rate is set as 20%, which means among a total number of 10 individuals in a one-time population, only the top two individuals will survive while all the others might die. As each generation of a population is a fixed number, eight new individuals will appear according to the natural selection theory.

- **Random Selection:** We also randomly select some of the individuals with low scores as the parents, because we want to promote genetic diversity. It is very likely that optimization algorithms get stuck at a local maximum. Consequently,

it may not reach the global maximum. By including some individuals who are not performing well, we decrease our likelihood of getting stuck. The random selection threshold is set to 0.1. Whenever a random number in the range of [0, 1] is generated, the number is compared with the probability threshold (e.g., 0.1). If the random number is larger than the threshold, then the corresponding individual will be temporarily kept in the gene pool. For all the genes with low fitness scores in the current population, a random selection procedure will be used to determine whether we keep the current individual or not.

- **Mutation:** Finally, mutation happens in a small random portion of the population which randomly modifies each individual. Like random selection, it also aims to encourage genetic diversity and avoids getting stuck at local maximum. As we only keep very few good individuals in each generation, we want to set the mutation probability considerably higher to speed up the revolution. For each individual in our genetic code, there are four possible positions that mutation might happen. For each individual, we restrict the operation to only change one position in one individual with a probability of 0.2. The position is also determined by a random integer ranging from [0,3]. The mutation will only flip the selected digit either from 1 to 0 or from 0 to 1.

- **Crossover:** If there are still empty slots left after retainment and random selection, a crossover will happen and fill out all the left portions. After reproduction of a new generation, the simple crossover may proceed in two steps. First step is the random pair selection. Second, each pair undergoes the cross over operation as follows: We take the first half digits from the male and the last half digits from the female. It is possible to have one parent breed multiple times, but the male and female parents cannot be identical. If the revolution process reaches a single optimal solution in an early stage, there will be no candidates remaining to process the crossover, since

both female and male will be always the same. Then, the new generation will stop with fewer individuals, as we only care about the top gene at the final stage.

- **Selection of the Best Individual:** With 20% survival rate (plus an additional 10% of other individuals) and 20% mutation, the evolution always takes less than three generations (10 individuals in each generation) to reach a perfect solution. If there are more than one individual in the last generation reaching to the same top fitness score, a logic "AND" operation is applied to those codes. For example, if both codes "1101" and "1100" perform the same, we will only use the features from the first two models instead of using three models, because the redundant features increase the computational complexity without a guarantee of boosting the performance.

4.4.2 Deep Representation Learning

After the genetic code evolution, we determined a varied number of deep features. It is difficult to determine a general network works for all of the combinations. Based on our experience, the feature set that contains less than 10k features could be translated into 256-dimension high-level feature neurons. Two optimizers are our candidates, namely Adam [173] and RMSprop [174]. Adam is used as the optimizer for the balanced datasets. If it is an imbalanced dataset, RMSprop is used, and one 50% dropout layer is inserted before the last output layer (softmax layer). Dropout can significantly reduce the effect of overfitting. Batch size is automatically determined according to the size of the training samples.

4.4.3 Experimental Analysis

- **Datasets:** We selected three representative datasets from different domains to evaluate our proposed idea. First, a Youtube video dataset [175] was used that repre-

sented different disaster event-related concepts and we extracted one keyframe from each video clip. Also, two image datasets were utilized for the performance evaluation. One was collected from the network cameras located in different places [176], while the other one was a well-known public dataset called CIFAR-10 [87] that classifies objects and animals.

For the datasets that are not separated into training and testing, we randomly select 20% of samples as testing and 80% as training. Specifically, Disaster video dataset was separated into training and testing based on the time the event happens (hurricane Harvey for training and hurricane Irma for testing). CIFAR-10 data already provided the training and testing data (50K for training, 10K for testing). Form the training dataset, 20% of the samples were selected as the validation set which calculated the fitness scores in the genetic code evolution to evaluate the genetic code. Also, the same validation data were used in the last stage to evaluate the performance during feature representation learning.

The statistical information of the first two datasets were listed in Table 4.5. Both of them are imbalanced datasets. For instance, the majority class in the Network Camera 10K is "Highway". In the Disaster dataset, the concept "flood/storm" contains most of the instances in both hurricane events. The list of concepts in CIFAR-10 dataset also shows in Table 4.6. It is a balanced dataset as each concept contains the same number of instances.

- **Experimental Setups:** The proposed framework was compared with several successful deep learning models proposed in recent years. More specifically, we selected MobileNet, ResNet50, Inception-v3, and VGG16 which are all pre-trained on the ImageNet dataset. We used linear SVM as the classifier for all the models.

Table 4.5: The statistical information of Network Camera 10K and disaster dataset

Network Camera 10K						Disaster			
No.	**Concepts**	**Instances**	**No.**	**Concepts**	**Instances**	**No.**	**Concepts**	**Harvey**	**Irma**
1	Intersection	855	**8**	Yard	161	**1**	Demonstration	42	8
2	Sky	495	**9**	Forest	139	**2**	Emergency Response	81	20
3	Water Front	978	**10**	Street	431	**3**	Flood and Storm	426	177
4	Building+Street	603	**11**	Parking	99	**4**	Human Relief	70	1
5	Park	499	**12**	Building	243	**5**	Damage	42	172
6	Montain View	719	**13**	Highway	3724	**6**	Victim	75	16
7	City	432	**14**	Park+Building	149	**7**	Speak	347	63
Total			**9527**			**Total**		**1083**	**457**

Table 4.6: Concepts in CIFAR-10 Dataset

No.	**1**	**2**	**3**	**4**	**5**
Concepts	Airplane	Automobile	Bird	Cat	Deer
No.	**6**	**7**	**8**	**9**	**10**
Concepts	Dog	Frog	Horse	Ship	Truck

In addition, we showed the performance of our genetic selection combined with a linear SVM and compared it with the whole proposed framework.

The evaluation metrics used in this work include Precision, Recall, average F1-score (Avg. F1), and weighted average F1 score (AvgW. F1) which take both imbalanced and balanced datasets into account. In particular, AvgW. F1 is calculated as the weighted sum of all F1 scores that considers the number of true instances for each class. This metric is important to show the performance of each model in a multi-class classification task.

Since Disaster dataset includes only 1K training data, the batch size of training model is set as 16, while the other two datasets are trained with batch size equals to 64. Since the proposed model only contains two dense layers at the end, we set the total number of epochs to 60 and only the best model with the lowest losses will be selected. As CIFAR-10 has larger amount of data compared to the other two datasets, the total number of epochs is set to 1200 to generate better training weights.

- **Experimental Results:** Table 4.7 illustrates the performance results for all the baselines as well as our proposed genetic selection and the whole framework. As it can be inferred from this table, ResNet50 usually performs better than other deep learning models such as VGG16, and Inception-v3 in imbalanced datasets (e.g., Disaster and Network Cameras). However, Inception-v3 can significantly improve the results compared to the ResNet50 in a balanced dataset like CIFAR-10. Overall, VGG16 performs poorly in all the selected datasets. MobileNet's results are very close to the ones from ResNet-50, which shows the effectiveness of this light-version model compared to the computationally-heavy models such as Inception-v3. These results show the necessity of an automatic model to select the best model or combine the best ones in a way to maximize the final classification results.

  Our proposed procedure of genetic code evolution shows the capability of identifying the best model or, in some cases, a group of models to further improve the final classification results. For the Disaster dataset, the best result is identical to ResNet50, which means combining any two or more of the models together will not improve the results. However, for the other two datasets, we can observe an improvement by combining several group of features from different pre-trained models. The AvgW. F1 improves 6% and 8% for Network Camera 10K and CIFAR-10, respectively.

  Finally, the whole framework shows more astounding improvements. We leverage the features by feeding them into another adaptive network instead of just simply concatenating them from each selected model. It can further improve the results considering all the evaluation metrics. Specifically, the AvgW. F1 reaches to 80% in the Network Camera 10K dataset. Furthermore, larger improvements can be recognized in the other two datasets.

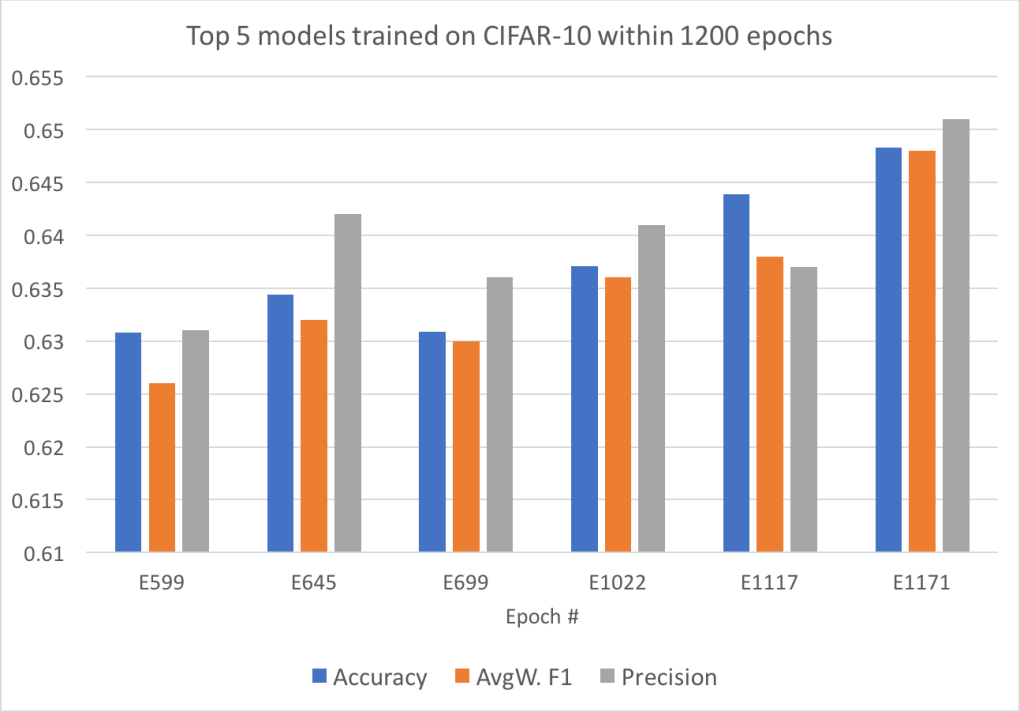


Figure 4.8: Top 5 models trained on CIFAR-10 within 1200 epochs

For CIFAR-10 dataset, we also visualize the accuracy (total number of correctly classified instances), AvgW. F1, and Precision from the last five models that were automatically saved during the training process with descending losses. From Figure 4.8, we can conclude that, although the Precision fluctuated over the time, a trend of further improvement can be expected with more iterations.

4.4.4 Conclusion

Currently, there exist many manually designed deep learning models which are successfully applied to different tasks. However, there is no automatic way to select the best model for each dataset and domain. To address this challenge, we propose a new genetic algorithm for deep learning optimization and model selection. Specifically, the proposed genetic encoding and the adaptive network can automatically select the best model

Table 4.7: Evaluation results on three different datasets using genetic selection algorithm with adaptive networks

Datasets	Models	Precision	Recall	AvgW. F1	Avg. F1
Disaster	MobileNet	0.260	0.092	0.380	0.121
	VGG16	0.140	0.142	0.296	0.109
	ResNet50	0.296	0.113	0.419	0.141
	Inception-v3	0.197	0.071	0.303	0.092
	Genetic Selection + Linear SVM	0.296	0.113	0.419	0.141
	Proposed Framework	**0.380**	**0.136**	**0.468**	**0.163**
Network Camera 10K	MobileNet	0.610	0.145	0.755	0.216
	VGG16	0.361	0.082	0.489	0.098
	ResNet50	0.640	0.158	0.773	0.233
	Inception-v3	0.559	0.131	0.726	0.194
	Genetic Selection + Linear SVM	0.668	0.174	0.797	0.254
	Proposed Framework	**0.700**	**0.182**	**0.804**	**0.261**
CIFAR-10	MobileNet	0.446	0.083	0.446	0.14
	VGG16	0.010	0.100	0.018	0.018
	ResNet50	0.471	0.09	0.469	0.15
	Inception-v3	0.502	0.102	0.503	0.169
	Genetic Selection + Linear SVM	0.588	0.137	0.589	0.223
	Proposed Framework	**0.651**	**0.169**	**0.648**	**0.268**

from the population. The experimental results show the effectiveness of the proposed GA method compared to other baselines.

4.5 Genetic Algorithm based Deep Learning Model Selection for Visual Data Classification

Classical ML techniques have achieved superior performance in many research domains for decades. Data modeling becomes challenging because uncertainties increase when applying ML to a broader area of study. Over the past several years, DL has overcame some of the limitations faced by classical ML for many research domains including visual data processing [177], speech recognition [178], and natural language processing [179]. Furthermore, traditional ML could not reach the same accuracy as the DL models in some cases, but developing and training a DL model from scratch is not always feasible

for all researchers with limited access to computational facilities. Usually, training a robust deep neural network is a computationally expensive task that requires high-end Graphics Processing Units (GPUs) to perform the training process in a reasonable time. Moreover, recent work indicates that not all neurons are needed after the completion of the first iteration. This surprising result may lead to the constructive definition of the wiring pattern, which today is weighted thrroug backprop and GAs. Fortunately, DL techniques are adaptable and transferable among different domains and applications. The rise in popularity of an optimization technique known as transfer learning [133] gave DL techniques the capability of influencing more scientific research areas and solving their domain-specific problems. Practical usage of the features, generated from well-designed pre-trained DL models, has enhanced the performance of many applications. Those models are not only transferable to similar domains, but also adaptable to different application fields. For example, the basic knowledge gained from a speech recognition task can now be easily applied to tasks in natural language processing [130].

Primarily, CNNs were intended to be utilized for basic image recognition, which resulted in a standout amongst the most well-known and broadly utilized deep learning methods. Different from traditional ANN models such as Multiple Layer Perceptrons (MLPs), which isolate the feature layers completely, CNN models take the raw picture as input with a two-dimensional structure and share the feature weights among local neuron connections. This change significantly reduces the number of parameters and makes the model simpler and easier to learn. Many CNN models are built and trained on ImageNet, a large scale public image dataset, and can be utilized in transfer learning to tackle visual data classification tasks in a broader target domain. Inception V3 [47] is an updated version from GoogleNet, which has the convolutional layers and pooling layers of the network separated in parallel. ResNet [53] overcomes the potential overfitting and vanishing gradient issue due to the increase of depth of the model, by constructing residual mod-

ules. MobileNet [54] is an efficient light weight CNN model for mobile and embedded vision applications. The standard convolutions are factorized into pointwise convolutions and depthwise convolutions. DenseNet, proposed by Huang et al. in 2016 [55], connects every layers to each other layers in a feedforward fashion. This modification obtains significant improvement by strengthening the feature propagation and encouraging the feature reuse, which substantially reduces the number of parameters.

There has been an increase in demand for automated optimization technology across the various industries in the world of business and technology. The search algorithms have a high capacity when it comes to delivering better designs within a short period of time. Choosing the most efficient optimization/search algorithm for a particular problem is dependent on the already defined design space. Some of the available algorithms include genetic algorithm, evolutionary programming, grid search, random search, and Bayesian optimization. The availability of those algorithms has enhanced performance across a wide range of problems, which eliminates the need for manual tuning.

To the best of our knowledge, there is no literature currently focusing on automatically determining the pre-trained deep learning model which fits a specific target domain. However, there are some optimization/search algorithms worth considering to tackle this problem. In this work, we proposed a Genetic Algorithm (GA) based deep learning model selection framework on identifying the feature set from a pre-trained model automatically. This feature set contains the most representative features of a specific dataset that could potentially improve the model's performance. This generalized framework can accommodate different datasets and problem domains. By integrating a two-stage genetic code evolution process, the proposed approach identifies the best feature layer or the layers' combination for a specific task to further build an image classification model.

Figure 4.9 illustrates the overall structure of the proposed automated model selection framework. The process for each candidate pre-trained deep learning model is indepen-

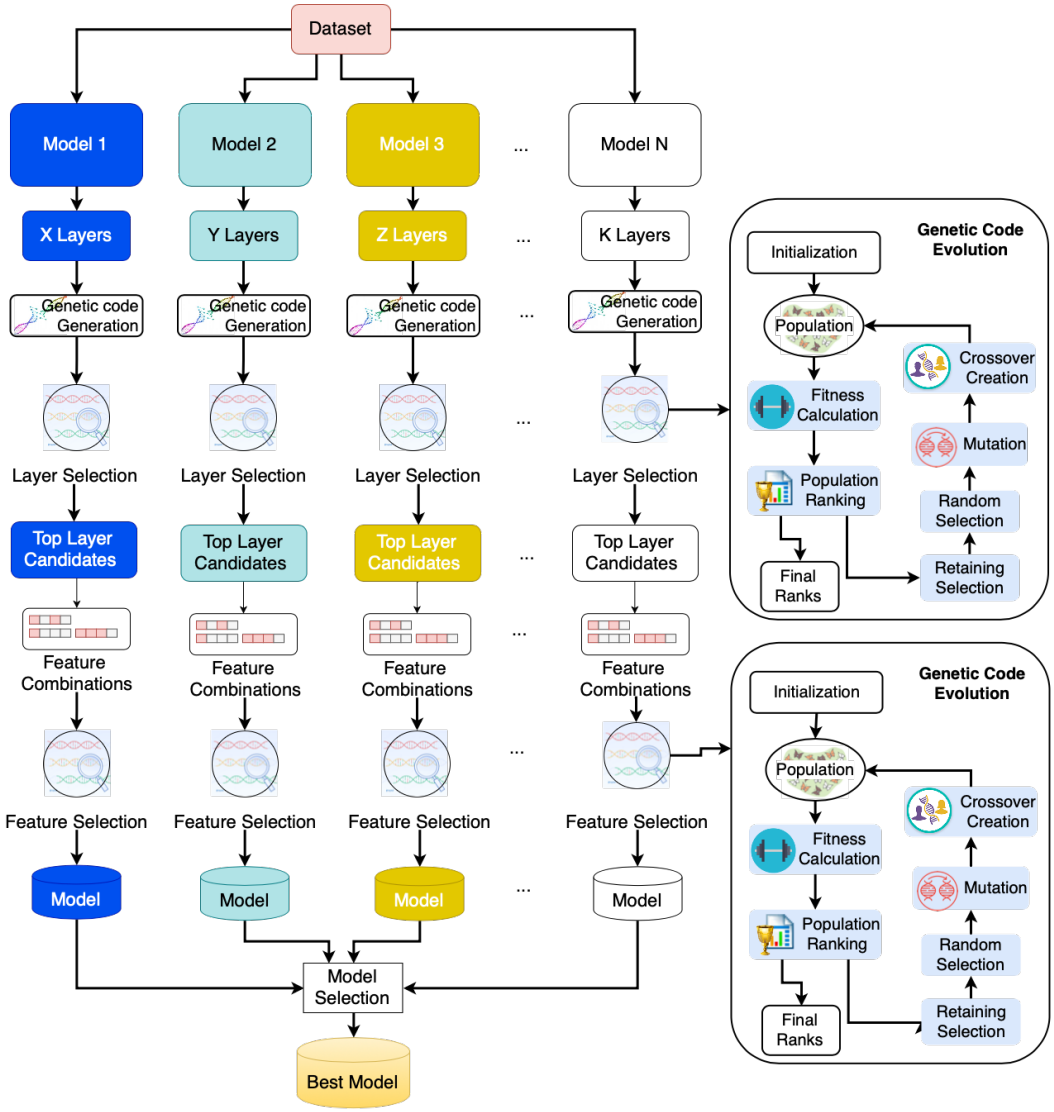


Figure 4.9: Proposed framework for deep learning model selection using a genetic algorithm

dent and can be run in parallel. Therefore, the processing time of the framework will not be significantly affected when adding more models to the comparison. This framework consists of two genetic code evolution processes to determine the best feature set for a specific input dataset. First, the top feature layers from each candidate model are selected in the layer selection phase. For each model $(1, 2, ...N)$, the number of layers that we extract features from will change $(X, Y, ...K$ layers). Therefore, the genetic code

generation accommodates the encoding of each feature layer as an individual with a fixed length considering the maximum number for each model. Then, the best feature combination is evaluated during the feature selection phase to generate the final features. This time, the encoding strategy changes to represent different combinations of the top layers. During the genetic code evolution process, several genetic operations are used to improve the average performance in each population. Each model's performance is validated in parallel using the best feature set. Only the model that shows the best performance on the validation data will be selected as the best model at the end.

A detailed explanation of the proposed framework is described next.

4.5.1 Genetic Code Evolution

Both the layer selection and feature selection phases use the same strategy to evolve the individuals, as shown in Algorithm 2. The initialization process randomly selects a certain number of individuals (we set it to 10 individuals empirically) and calculates the fitness score for each one of them. The fitness score is generated by the fitness function $f(i)$ (line 5), which is the average F1 score (Avg. F1):

$$f(i) = (\sum_{c=1}^{C} \frac{2 * P_c^i * R_c^i}{P_c^i + R_c^i})/C, \tag{4.13}$$

where C is the total number of classes in the target dataset; i is a unique individual; P is the precision of class c, and R is defined as the recall of class c. Precision represents the classifier's ability to not label a positive sample as negative, while recall represents the classifier's ability to find all the positive samples. The relative contribution of precision and recall to the F1 score is equal, which makes it a trade-off between these two evaluation criteria.

The individuals in a specific population are ranked in descending order to create a ranking list. Based on a predefined retention, the individuals on the top of the list will

Algorithm 2: Genetic Code Evolution

```
1  RETAIN ← 0.4
2  SELECT ← 0.1
3  MUTATE ← 0.2
4  for individual i ∈ Population p do
5      calculate FITNESS FUNCTION f(i)
6      grade[i].score ← f(i)
7  Sort grade in descending order
8  topGrade = grade[0 : RETAIN * grade.size]
9  restGrade = grade[RETAIN * grade.size : grade.size]
10 for x ∈ topGrade do
11     parents.append(x)
12 # Random selection
13 for x ∈ restGrade do
14     if SELECT > random() then
15         parents.append(x)
16 #Mutation
17 for x ∈ parents do
18     if MUTATE > random() then
19         MUTATE(x)
20 # Crossover
21 size ← Population.size − parents.size
22 while children.size < size do
23     select famale and male randomly from parents
24     if female ≠ male then
25         child = (male.partA + female.partB)
26         children.append(child)
27 parents.append(children)
28 return parents
```

continue to the next generation and will produce offspring. Other individuals will have a small chance to survive depending on the random selection process in lines 12 - 15. All other individuals will be discarded for the next generation.

Figure 4.10 and 4.11 depict the genetic code evolution process for one generation for the layer selection and feature selection phases respectively. The figures illustrate the process of the mutation and crossover operations referring to lines 16 - 19 and lines 20 - 27 in the algorithm.

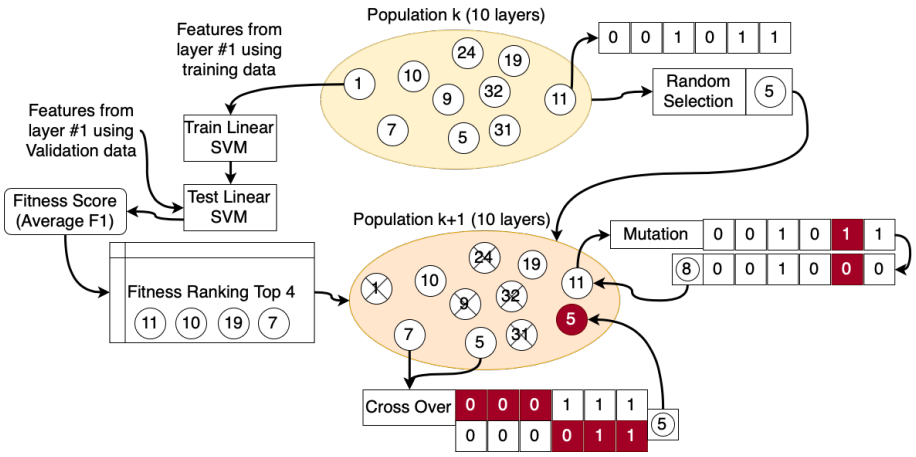


Figure 4.10: Genetic code evolution example for one generation in the layer selection phase

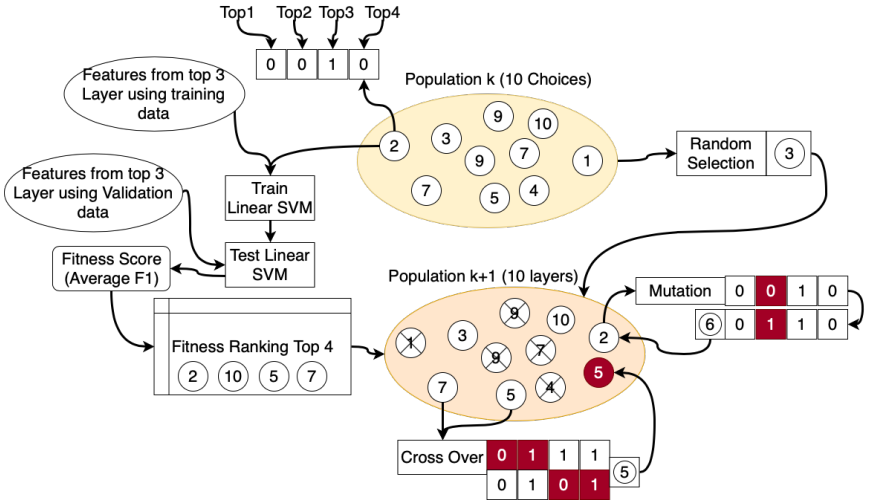


Figure 4.11: Genetic code evolution example for one generation in the feature selection phase

4.5.2 Layer Selection Phase

In the layer selection phase, genetic encoding operation transforms the ID of each feature layer into a unique binary string. The available layers for feature extraction in each model are different, which makes the corresponding encoding bits different for each model's process. For each individual, the features from a particular layer are extracted using the training data to build a Linear SVM classifier. Furthermore, the features from the same

layer are obtained using the validation data to evaluate the classifier's performance and calculate the corresponding individual's fitness score.

A one-digit change (0 to 1 or 1 to 0) in the evolution process will result in choosing a different layer for feature extraction, affecting the performance of the classifier. For instance, changing 001011 to 001001 means that layer #8 will be selected instead of layer #11 to generate the features. This operation applies to the mutation process, where we restricted the process to affect only one position of the encoding each time for one selected individual.

The crossover operation generates new individuals for the next population by combining the genetic codes from two retained individuals. Each parent contributes only the left half or the right half of the genes. A new individual is then added to the next population by combining these two halves to create a new genetic code of the same length. For example, taking the left half of the genetic code 000111 and the right half of the genetic code 000011 will generate another individual represented by the genetic code 000011.

4.5.3 Feature Selection Phase

After evolving the individuals in the layer selection phase for several generations, the last generation identifies the top layer candidates to represent the most reliable features for a specific dataset. The best individuals are determined by the predefined retention rate from the final ranking list. Those features will be further encoded as different feature combinations to proceed with feature selection.

Different from the previous stage, where each genetic code represents a single feature layer, in the feature selection phase each binary string encodes a way of combining features from different layer candidates. The mutation and crossover operations as described in the previous section remain the same, except that each digit means a top feature set

Table 4.8: The pre-trained deep learning model candidates with the available number of feature choices

Models	Layers	# combinations
InceptionV3	94	94^4
ResNet50	64	64^4
MobileNet	13	13^4
DenseNet201	80	80^4
Total combinations	–	$3.41E32$

will be selected or deselected to form the final feature set (e.g., a "0" means do not select, while a "1" means select). Therefore, a mutation process will either add a new feature layer or remove a feature layer from the final feature set.

After finishing the second phase of the genetic code evolution process, we selected the top feature set from each pre-trained deep learning model with the highest average F1 score running on a Linear SVM classifier. The final model is determined by comparing the average F1 scores using the same validation data to extract features from each model. The model with the highest score will be selected as the best feature extractor to build the final classification model.

4.5.4 Experimental Analysis

Experimental Setup

We chose four pre-trained deep learning models as our model candidates. The model and the corresponding number of available layers are shown in Table 4.8. As we set each population to generate 10 individuals, and the retention rate (r) to 0.4, for each model's layer selection phase we have a total number of K^{10*r} feature set choices, where K is the available number of layers for each model. As the final output is limited to the feature set from one model, the total number of possible choices to determine an optimal solution adds up to $3.41E32$. The space is far too large to be explored exhaustively by hand.

We used four datasets from different domains to evaluate our proposed approach: two imbalanced and two balanced datasets. One of the imbalanced datasets is a disaster video dataset that consists of two major hurricane events that happened in 2017 in two different geographic locations: Harvey in Texas and Irma in Florida. The other imbalanced dataset is a surveillance camera dataset that contains images captured from a variety of places. Table 4.9 shows the statistical information of these two datasets. In the Disaster dataset, the "Flood and Storm" concept contains most of the samples. For the disaster dataset, by following a chronological order, we use the first event as the training data and the second event as the testing data. We extracted one representative keyframe image for each video. For the Network Camera 10K dataset, 20 percent of the data was separated into testing data. Moreover, 20 percent of the training data from both datasets was randomly selected to form the validation data for the fitness score calculation. The majority class in this dataset is the concept "Highway". The two balanced datasets (CIFAR10 and MNIST-Fashion) are well-known public datasets. CIFAR10 classifies objects and animals, and MNIST-Fashion serves as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. These two datasets were already split into training and testing, but we randomly selected 20% of the training samples for our validation data to calculate the fitness score during the genetic code evolution process. Both datasets consist of an equal number of samples for each class. CIFAR10 includes concepts related to objects (e.g., airplane, automobile, ship, and truck) and animals (e.g., bird, cat, deer, dog, frog, and horse). MNIST-Fashion is a collection of grayscale images of clothing types such as t-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot.

We compared the performance of our proposed framework with that of the three optimization algorithms mentioned in the related work. Each of those algorithms selects a pre-trained model as the best feature model. Bayesian Optimization is chosen to evalu-

Table 4.9: The statistical information of the Network Camera 10K and Disaster dataset

Network Camera 10K						Disaster			
No.	**Concepts**	**Instances**	**No.**	**Concepts**	**Instances**	**No.**	**Concepts**	**Harvey**	**Irma**
1	Intersection	855	**8**	Yard	161	**1**	Demonstration	42	8
2	Sky	495	**9**	Forest	139	**2**	Emergency Response	81	20
3	Water Front	978	**10**	Street	431	**3**	Flood and Storm	426	177
4	Building+Street	603	**11**	Parking	99	**4**	Human Relief	70	1
5	Park	499	**12**	Building	243	**5**	Damage	42	172
6	Montain View	719	**13**	Highway	3724	**6**	Victim	75	16
7	City	432	**14**	Park+Building	149	**7**	Speak	347	63
Total			**9527**			**Total**		**1083**	**457**

Table 4.10: Proposed framework's final model performance on four datasets compare to Bayesian optimization, evolutionary programming, and genetic algorithm without mutation operation

Datasets	**Algorithms**	**Final Model**	**Precision**	**Recall**	**Avg. F1**	**W. Avg. F1**
Disaster	Bayesian Optimization	InceptionV3	0.3215	0.3256	0.2747	0.3920
	Evolutionary Programming	InceptionV3	0.3192	0.3084	0.2514	0.3937
	Genetic Algorithm w/o mutation	InceptionV3	0.3215	0.3256	0.2747	0.3920
	Proposed Method	**ResNet50**	0.3212	**0.3276**	**0.2867**	**0.4430**
Network Camera 10K	Bayesian Optimization	ResNet50	0.6398	0.6263	0.6290	0.7827
	Evolutionary Programming	InceptionV3	0.6644	0.5896	0.6108	0.7705
	Genetic Algorithm w/o mutation	ResNet50	0.6391	0.6081	0.6175	0.7761
	Proposed Method	**InceptionV3**	0.6508	**0.6339**	**0.6409**	**0.7985**
CIFAR10	Bayesian Optimization	ResNet50	0.8949	0.8943	0.8945	-
	Evolutionary Programming	ResNet50	0.8996	0.8995	0.8995	-
	Genetic Algorithm w/o mutation	ResNet50	0.8934	0.8928	0.8930	-
	Proposed Method	**ResNet50**	**0.9063**	**0.9061**	**0.9061**	-
MNIST -Fashion	Bayesian Optimization	ResNet50	0.9260	0.9263	0.9260	-
	Evolutionary Programming	ResNet50	0.9282	0.9285	0.9282	-
	Genetic Algorithm w/o mutation	ResNet50	0.9282	0.9285	0.9282	-
	Proposed Method	**ResNet50**	**0.9289**	**0.9292**	**0.9289**	-

ate if its advantage regarding probability assumptions will have a positive impact on our specific task. Evolution programming and genetic algorithm without mutation operations are included in the comparison to determine whether or not both mutation and crossover operations are necessary to converge to the optimal solution.

Experimental Results

The performance of the proposed framework compared to the other three optimization algorithms on the four datasets and with different pre-trained models is shown in Ta-

ble 4.10. Four metrics were considered: Precision, Recall, averaged F1 scores [Avg. F1], and Weighted average F1 score [W. Avg. F1]. For evaluating the performance of an imbalanced dataset, the F1 score is a a better measure than accuracy. As the F1 score captures the trade-off between precision and recall, it is more suitable to evaluate the overall model performance. In this framework, we use a Linear SVM model to evaluate the feature performance on the validation data. Therefore, the reported evaluation metrics are based on the testing results of the SVM classifier's output.

As shown in Table 4.10, our proposed framework always selected the pre-trained model with the best performance on each dataset. For the disaster dataset, only our proposed method selected ResNet50 to extract the feature set. The performance improved more than 5% compared to the others using W. Avg. F1. For Network Camera 10K dataset, though Evolutionary Programming selected the same model as our proposed method, the overall performance is the worst, which means it failed to identify the best feature set. The other two models selected ResNet50 to produce the final feature set, while our method select InceptionV3 and has a better performance on the testing.

For the balanced public datasets, CIFAR 10 and MNIST-Fashion, all methods selected ResNet50 as the best feature model. We didn't report W. Avg. F1 scores here because they are identical to the Avg. F1 scores when each class has the same number of samples. Though all the methods selected the same pre-trained model, the features performances are not all the same. Our method is the only one can bring CIFAR 10 data's performance beyond 90%. Though all the methods' performance are very close in MNIST-Fashion, our method still identifies the feature set with the best performance.

Figure 4.12 - 4.15 illustrate the single model's feature performance using the proposed model and the other three optimization algorithms. The y-axis in all the figures represents the evaluation metrics' scores (ranging between 0 to 1). The purpose of these comparisons here is to ensure that our proposed method could always determine the best feature set for

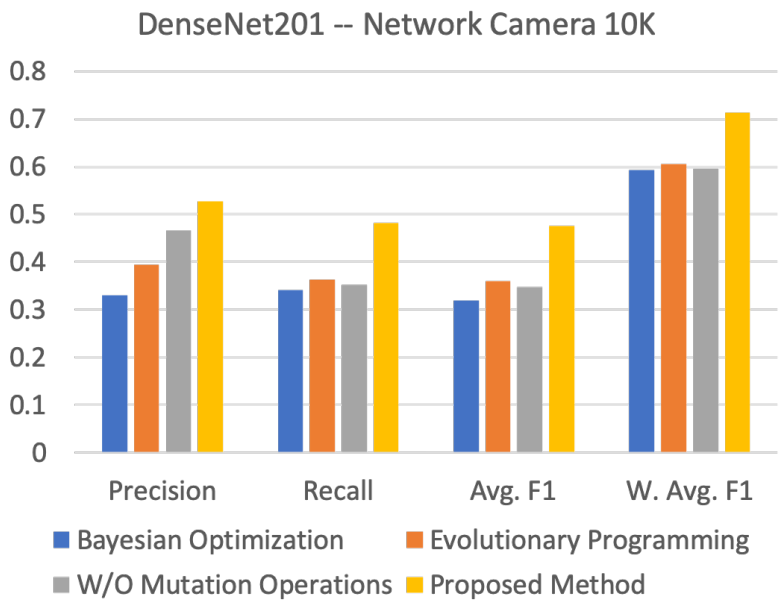


Figure 4.12: DenseNet201 model performance on Network Camera 10K dataset

a specific dataset no matter how the candidate models change. Though DenseNet201 and MobilenNet models are not selected by any one of the optimization algorithms for the four experimental datasets, from the bar chart we can tell, the proposed method always have the best performance consider all the evaluation metrics. Figure 4.15 also shows that the proposed method identifies the best feature set from InceptionV3 model, however it does not select it as the best model. Thus, it is true that this model's best feature set's performance cannot compete with the feature set from ResNet50 model as we showed in the experimental result table.

4.5.5 Conclusion

We identify the potential challenges of using pre-trained deep learning models on different target problem domains. We proposed to build a generalized framework using genetic algorithms to automatically determine the best feature set from a group of model candidates. A feature set that contains the most representative features for a specific target

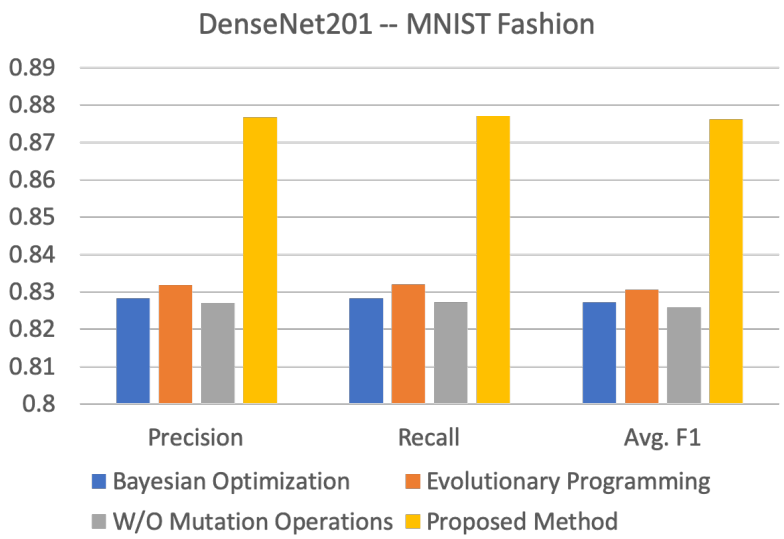


Figure 4.13: DenseNet201 model performance on MNIST-Fasion dataset

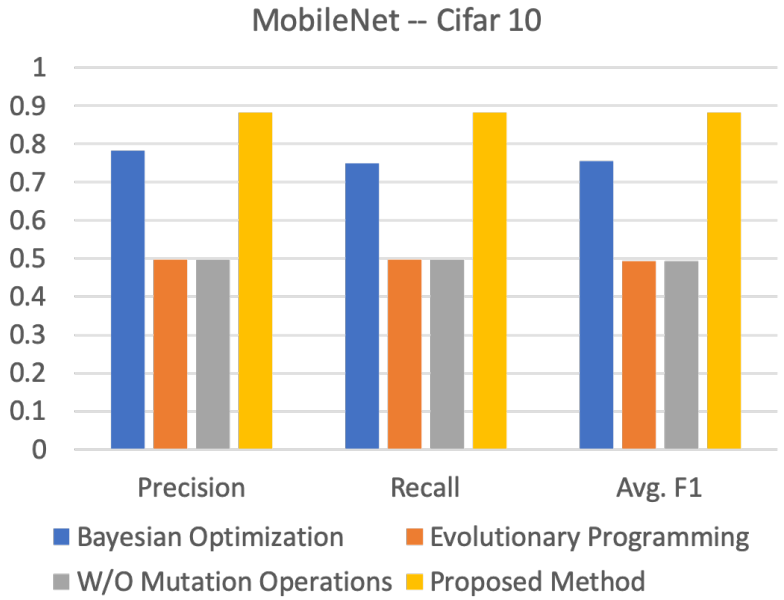


Figure 4.14: MobileNet model performance on CIFAR10 dataset

domain can be better utilized to train a classifier, then further enhance the final model's performance. The experimental results have shown that our proposed approach outperformed the other optimization algorithms and can always identify the best feature set no matter how the model candidates change. Since each model candidate is processed

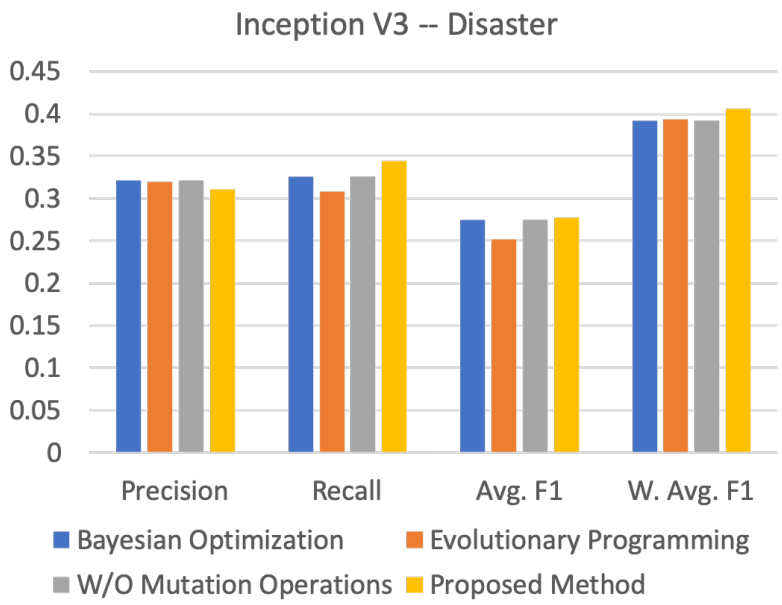


Figure 4.15: InceptionV3 model performance on Disaster Dataset

and evaluated independently, the framework can be run in parallel and makes the time-consuming task to be more efficient.

4.6 Automated Neural Network Construction with Similarity Sensitive Evolutionary Algorithms

Through the transfer learning process, traditional ML techniques can directly use the high-level semantic features learned from the DL models to perform many other domain-specific tasks while achieving higher performance than before. At the same time, constructing another comparatively simple deep neural network as the base network also appears to be a popular choice. When the new application domain is not very close to the source domain, deep features need to be transferred to better represent the targeted application. Still, even when adding just one layer to the pre-trained model, there is no guarantee that the new model will get promising results as expected in the source domain.

Although we benefit in transfer learning, it still takes time to design a neural network after the feature extraction process – especially when the researcher has limited experience or knowledge of neural networks, the datasets, and the target tasks. To tackle this emerging issue, studies have recently focused on automating the network design process [180, 181]. Two favorable directions to further explore are reinforcement learning [182] and Evolutionary Algorithms (EA) [183]. The latter has shown great promise in solving complex problems that the former has defined for several decades [184, 185, 186].

In this work, we aim to leverage the deep features from pre-trained Convolutional Neural Network (CNN) models in different applications without the need to spend most of the effort on examining the characteristics of each task. We propose a generalized framework to accommodate different datasets and problem domains. By integrating EA and other techniques to support the automated searching process, the hyperparameters of a new neural network built for a specific task are determined after the best individual is selected.

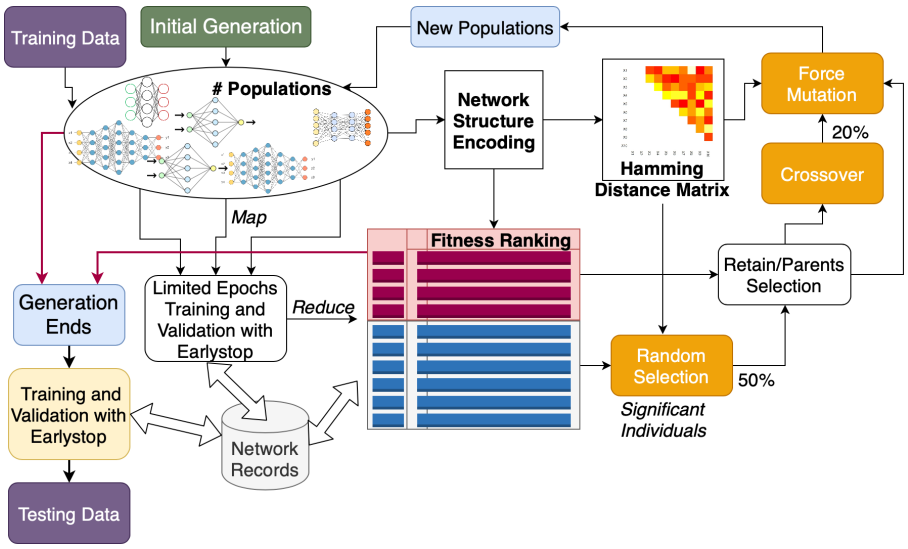


Figure 4.16: Proposed framework for automated neural network construction

An EA-based framework for automated neural network construction is illustrated in Figure 4.16. This framework aims to select the best network model for a specific task that

uses transfer learning for image classification. Four major hyperparameters (i.e., number of neurons in one layer, number of fully connected layers in one model, the activation function, and the optimizer) are considered to formulate the network's gene. Specifically, a combination of those four hyperparameters composes a unique gene sequence that represents a network. The search process starts by randomly generating a group of networks as the initial population. Then, the initial networks evolve for several generations until they reach the end of the evolutionary limit. Several evolutionary strategies are used in this framework to improve the average performance in each population. Meanwhile, a Hamming distance matrix is calculated for every generation to evaluate the structural differences between each individual. Different genetic operations will be taken as reactions to the similarity evaluation, which ensures that the development of the new generation continues to cover a large searching space. The model that performs the best on the validation data is then identified at the end of the network's evolution. Next, a complete training process starts to build the final model for the targeting task.

4.6.1 Network Selection

Network selection starts with an initial population that is generated by a random search. The process, as shown in Algorithm 3, takes all the networks (individuals) into the current generation to evolve. The network evolution incorporates all the genetic operations that might be triggered during the evolving process for every generation. The proposed evolutionary process enhances the operations in the traditional GA by controlling the similarities between the populations in subsequent generations. Specifically, it takes the strength of mutation operation in evolutionary programming to overcome the underlying weakness of crossover in the later generations. Force mutation and distance calculation ensure that the evolution process is capable of exceeding a local optimum in the searching

Algorithm 3: Network Evolution

1 $RETAIN \leftarrow 0.4$, $SELECT \leftarrow 0.5$, $MUTATE \leftarrow 0.2$
2 **for** *individual* $i \in$ *Population* p **do**
3 calculate FITNESS FUNCTION $f(i)$
4 $grade[i].score \leftarrow f(i)$
5 Sort $grade$ in descending order
6 **for** $u \in [0, grade.size - 1]$ **do**
7 $codes[u] \leftarrow$ENCODING ($grade[x].network$)
8 **for** $v \in [v + 1, grade.size - 1]$ **do**
9 $H_{uv} \leftarrow \sigma$ ($codes[u], codes[v]$)
10 $Significant =$ (MAX(H)$-$MIN(H))$/2$
11 **for** $x \in [0, RETAIN * grade.size - 1]$ **do**
12 $parents.append(grade[x])$
13 # Random selection
14 **for** $x \in [RETAIN * grade.size, grade.size - 1]$ **do**
15 **if** $SELECT > random()$
16 *AND* $\forall H_{xs} > Significant$ *WHERE* $s \in parents$ **then**
17 $parents.append(grade[x])$
18 # Crossover
19 $size \leftarrow Population.size - parents.size$
20 **while** $children.size < size$ **do**
21 select $famale$ and $male$ randomly from $parents$
22 **if** $female \neq male$ **then**
23 $child = (male.partA + female.partB)$
24 **if** $MUTATE > random()$ **then**
25 MUTATE($child$)
26 $children.append(child)$
27 **else**
28 # Force Mutation
29 $child = male$
30 MUTATE($child$)
31 $children.append(child)$
32 $parents.append(children)$
33 **return** $parents$

space when the top networks in the same generation are very similar. In each generation, a portion of the top networks is selected as the parents to produce offspring that represent the new network structures. The selection is based on a specific retaining rate and a fitness function ranks the networks (lines 4 - 7). This fitness function is based on the formula of

the averaged F1 score, which evaluates the performance of a specific network. Compared to using accuracy as the evaluation criterion, F1 score is more suitable for evaluating any dataset, whether balanced or imbalanced. The fitness function can be written as follows:

$$f(i) = (\sum_{c=1}^{C} \frac{2 * tP_c^i}{2 * tP_c^i + F_c^i})/C, \tag{4.14}$$

where C is the total number of classes in the targeting dataset, and i is the index of a unique network. tP is defined as the number of instances correctly identified as class c, and F is defined as the number of instances wrongly classified as c or others respectively for the network i.

Besides updating the ranking list of networks with their performance in each evolution, we use additional storage to record the information of the networks that have already appeared in previous generations. As we care only about the best performance that we have gotten for each unique gene sequence, only the highest fitness score for one combination of the hyperparameters will be stored for later references. Multiple networks that share the same gene, however, can appear in the ranking list, which represents the overall performance of the current population. The more times this structure is selected, the greater the chance that the offspring of the next generation will have substantial similarity between the compositions of each network's gene. To overcome this limitation, which might slow down the evolving process and keep the solution at a local optimal, distance calculation and force mutation play vital roles in ensuring that population can keep searching for more combinations in the later generations after identifying several network configurations with proper performance.

Genetic encoding (line 7) transforms one combination of four candidate hyperparameters into a unique binary string. Table 4.11 shows the available choices of each hyperparameter and the corresponding encoding bits. One-bit flipping (0 to 1) will result in changing of one hyperparameter, consequently affecting the performance of the network

Table 4.11: The available choices for network hyperparameters and the corresponding binary encoding digits

Hyper-parameters	**Choices**	**# Encoding Digits**
# neurons	32, 64, 128, 256, 512, 768, 1024	3
# layers	1, 2, 3, 4	2
Activation functions	relu, elu, tanh, sigmoid	2
optimizers	rmsprop, adam, SGD, adagrad,adadelta, adamax, nadam	3

(e.g., for the choice of # layers, 00 to 10 means adding two more layers) . The hamming distance calculation (line 9) is used to generate the distance matrices $H = (\sigma(x, y))$, where $1 \leq x \leq P$ and $1 \leq y \leq P$, and P is the designated number of individuals in one population. Lines 13 - 31 illustrate the procedure of all the genetic operations (conditional random selection, crossover, and force mutation) within specified activating conditions (defined in line 1). While doing the crossover operation, we select two network genes as the candidates of the parents. Each gene representing the combination of four hyperparameters is separated into two parts. Part A consists of the first two hyperparameters in the table, and part B takes the rest. By evenly separating one combination into two opposite parts, the proposed approach holds a lower bound p_s of the survival probabilities under simple crossover.

$$p_s = 1 - \sigma(H)/(l - 1), \tag{4.15}$$

where $\sigma(H)$ is the distance between the two digits we are observing, and l is the total length of the gene sequence (10 in this case). As the number of neurons in one layer can significantly affect the choice of the number of layers, grouping the choices of the first two hyperparameters onto one side of the cutting point will provide higher survival probabilities compare to separating them into two parts. As the mutation operator randomly

activates after crossover, the survival probabilities reduced to

$$p_s = 1 - \sigma(H)/(l-1) - (1-p_m)^{po}, \tag{4.16}$$

where p_m is the mutation probability. Instead of changing one bit of the genetic code, the mutation operator randomly chooses a value for one specific hyperparameter. Therefore, the maximum number of fixed position po is reduced from 10 to 4 (since we have 4 hyperparameters). By reducing po, the mutation effect increases ensuring a further decrease of the lower bound of the survival rate. We expected these changes could help on gene evolution process when most of the individuals in the later generations are very similar. Section 4.6.3 further details how this strategy met our expectations. Again, note that this evolutionary process, while illustrative is not heuristic or strong and thus will not scale successfully in its present form.

4.6.2 Network Construction and Training Process

After evolving the networks for a certain number of generations, the last generation identifies the top candidate to construct the final training model. As the output layer separates the data into different classes, the number of neurons in the layer right before the output layer should not greatly exceed the number of classes for each specific task. Therefore, the number of neurons that the genetic selection process determines sets the number for only the first layer. The number of neurons will gradually diminish by half until either it reaches the last fully connected layer or the number of neurons in the current layer is less than twice that of the number of classes. Also, one 50% dropout layer is placed ahead of the output layer to reduce the effect of overfitting.

Moreover, we restricted the training epoch in the network selection phase to a relatively small but reasonable number (200), so the converging speed of the network has been considered by default. Nevertheless, the early stopping with patience=5 is set for all

Table 4.12: The statistical information of the Network Camera 10K and disaster dataset

Network Camera 10K						Disaster			
No.	**Concepts**	**Instances**	**No.**	**Concepts**	**Instances**	**No.**	**Concepts**	**Harvey**	**Irma**
1	Intersection	855	**8**	Yard	161	**1**	Demonstration	42	8
2	Sky	495	**9**	Forest	139	**2**	Emergency Response	81	20
3	Water Front	978	**10**	Street	431	**3**	Flood and Storm	426	177
4	Building+Street	603	**11**	Parking	99	**4**	Human Relief	70	1
5	Park	499	**12**	Building	243	**5**	Damage	42	172
6	Montain View	719	**13**	Highway	3724	**6**	Victim	75	16
7	City	432	**14**	Park+Building	149	**7**	Speak	347	63
Total			**9527**			**Total**		**1083**	**457**

models, which means that most of the time the training process will not last until the last epoch. Ideally, we can identify a network that performs well in more generations during genetic selection within a competitive training duration.

4.6.3 Experimental Results

We evaluate the proposed framework using two datasets: a disaster video dataset that consists of two major hurricane events that happened in 2017 in two geographic locations (Harvey in Texas and Irma in Florida), and a surveillance camera dataset that contains images captured from various places. Table 4.12 shows the statistical information of these two datasets. For the disaster dataset, by following chronological order, we use the first event as the training data while the latter one becomes the testing data. We extract one keyframe image as the representative of each video. For the Network Camera 10K dataset, 20% of the data is separated into testing data. Moreover, 20% of the training data from both datasets was randomly selected to form the validation data for fitness score calculation which assists the model training and network evolution process.

Before getting into the evaluation of the final model, we observed the efficiency of the proposed framework. In Figure 4.17, the fitness scores' distribution (average F1 scores of the validation data) of the top 12 individuals in each generation depicts the evolutionary process. Since the retaining rate in the evolutionary process is defined as 0.4 and the

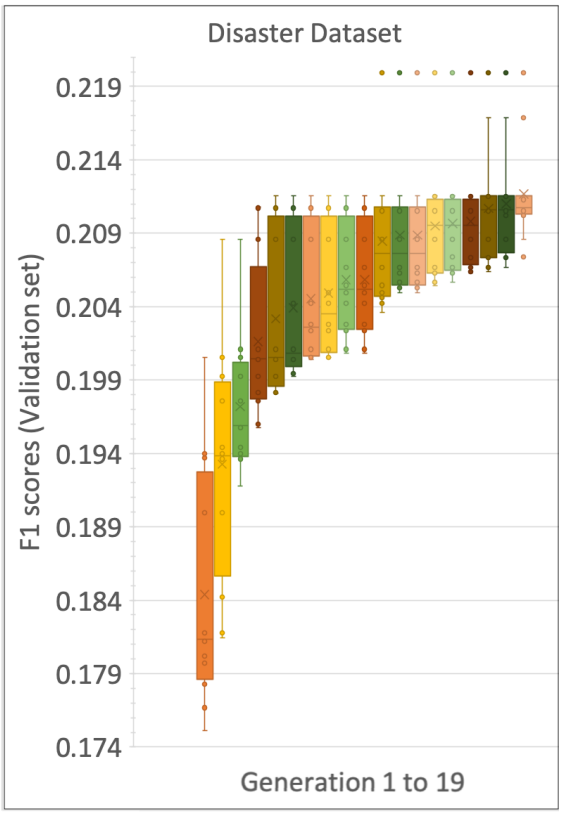


Figure 4.17: The performance of the top 40% of the individuals in each generation for the Disaster Dataset

number of populations in each generation is fixed at 30, only the top 12 individuals will survive and continue evolving in the next generation. As can be seen from the plot, the performance of each generation has steadily increased and reached a certain optimal F1 score near the 5th generation. After that, the new populations in each generation keep searching for a better solution and successfully exceed the optimal score in the 10th generation. Notably, the model not only focuses on discovering one individual as the best solution but also raises overall performance gradually for all the top populations in subsequent generations. Similar trends can be also found in Figure 4.18. Since we have a limited number of GPUs, we reduce the total number of populations in each generation to 25 for the Network Camera 10K dataset. As the retaining rate stays the same, the plot

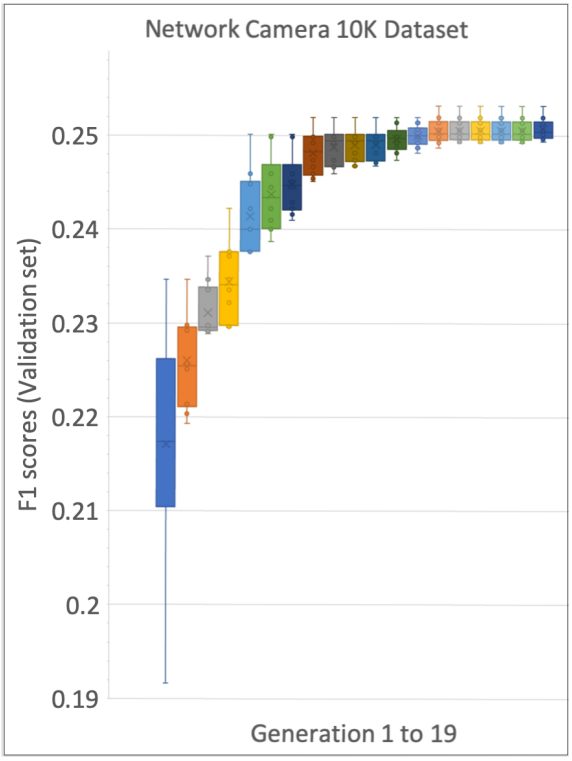


Figure 4.18: The performance of the top 40% of the individuals in each generation for the Network Camera 10K Dataset

shows the performance of the top 10 individuals for each generation. Those 10 populations in each generation are the parents that contribute to the next generation. Similarly, a local optimal appears near the 8th generation and sticks for several iterations. Again, our framework successfully gets the F1 score to improve after the 14th generation.

Furthermore, Figures 4.19 - 4.22 compare the search space covered in the representative generations and visualized all the individuals using scatterplots in three-dimensional space that clearly shows the improvement of the three generations (the first, the 10th, and the last generation). We project each unique network structure into a two-dimensional space, where the x-axis represents the gene code in decimals of the first two hyperparameters, and the y-axis represents the other two. Five binary digits can be easily converted to a decimal number between 0 and 31. Therefore, a unique pair of x and y (point $[x, y]$

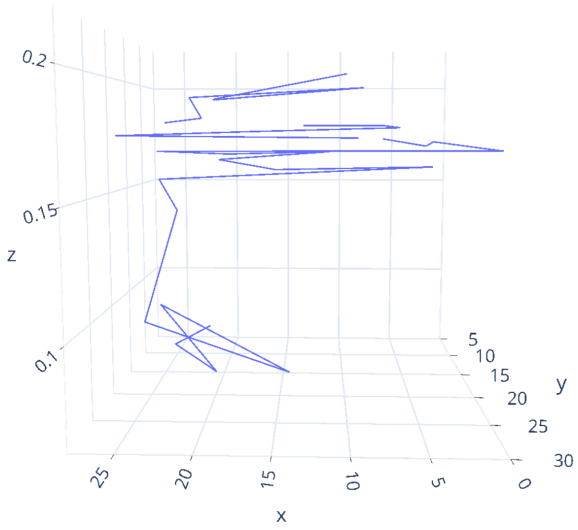


Figure 4.19: The first generation search

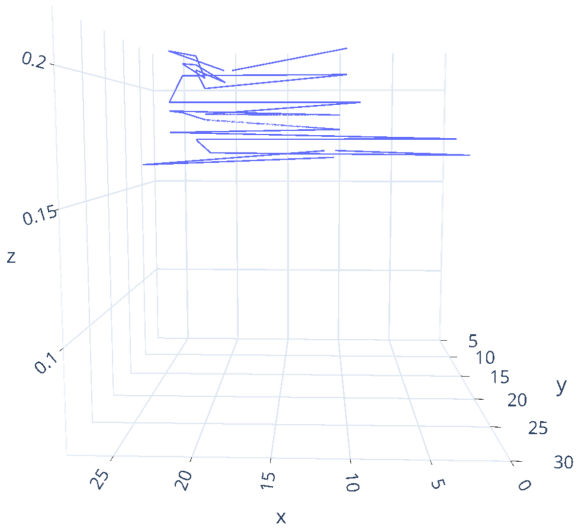


Figure 4.20: The tenth generation search

in the plots) represents a unique individual in the search space. The z-axis is the fitness score, which means the dots on the top represent the models that have better performance. The first generation starts with randomly selected individuals, covering a sparse space with various performance measures. Until the 10th generation, the individuals with lower scores are eliminated from the population. The solutions, however, are narrowed down into a smaller space. As we proposed a similarity sensitive framework, it still makes a

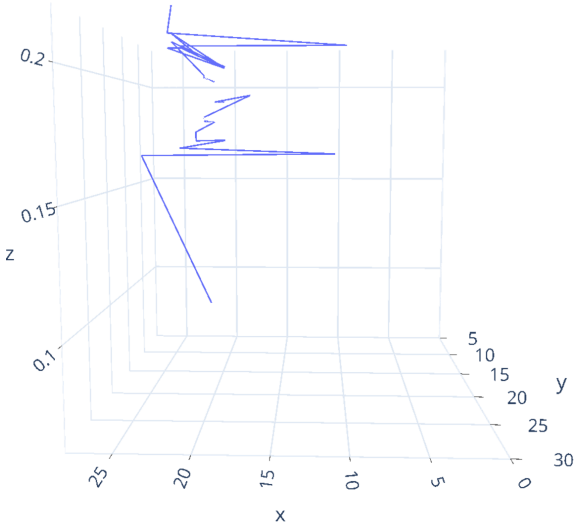


Figure 4.21: The last generation search

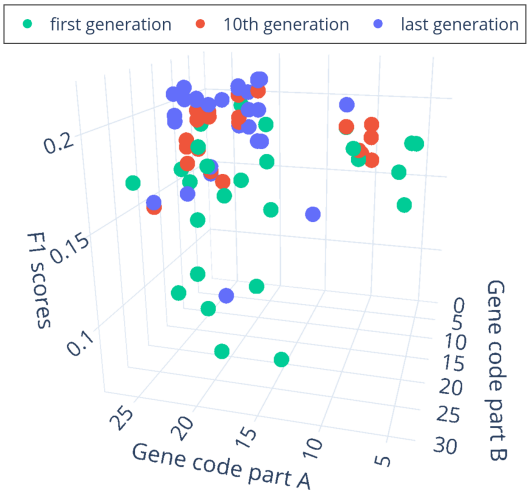


Figure 4.22: Individual performance in the first, tenth, and the last generation (Disaster)

breakthrough in the search space later on and produces better results. It is also evident in Figure 4.22 that the first generation (green dots) sparsely covers the search space by randomly producing 30 populations. Until the 10th generation (red dots), the individuals stay in a smaller solution area and demonstrate average performance. Finally, in the last generation (blue dots), the individuals became sparse again to cover a larger search space, which resulted in an optimal better than the local optimal than had been reached in the

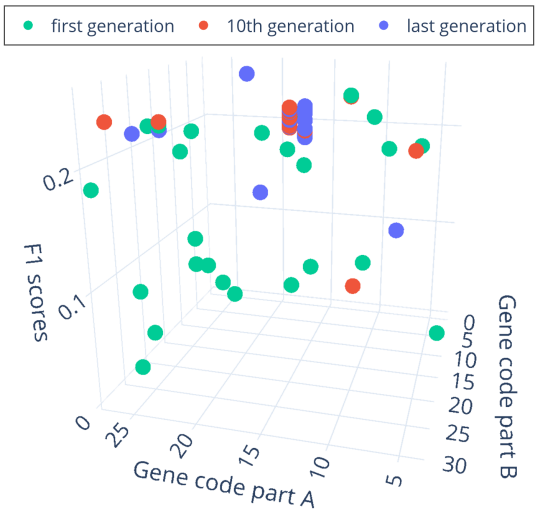


Figure 4.23: Individual performance in the first, tenth, and the last generation (Network Camera 10K)

very early stage. We also plot the performance of the model using the Network Camera 10K dataset in the same way as shown in Figure 4.23. In the 10th generation, the network candidates have been identified in a very restricted area. Still, the proposed framework shows the power of escaping the limited space and getting better solution to construct the network with extensive performance.

Table 4.13: Evaluation results on two datasets along with the final hyperparameters configurations

Datasets	**Models**	**AvgW. F1**	**Avg. F1**
Disaster	MobileNet	0.380	0.121
	ResNet50	0.419	0.141
	Inception-v3	0.303	0.092
	Our Work	**0.541**	**0.194**
Hyperparameters: 768 Neurons, 2 Layers, sigmoid, adamax			
Network Camera 10K	MobileNet	0.755	0.216
	ResNet50	0.773	0.233
	Inception-v3	0.726	0.194
	Our Work	**0.806**	**0.268**
Hyperparameters: 256 Neurons, 1 Layer, sigmoid, adamax			

The overall performance of the proposed framework is listed in Table 5.4 and compared with different pre-train models using two criteria (weighted average F1 score [AvgW. F1], and averaged F1 scores [Avg. F1], respectively). For evaluating the performance of an imbalanced dataset, F1 score is a more vital measure than accuracy. As trade-offs between precision and recall, F1 scores are more suitable to evaluate the overall model performance.

Generally, using a pre-trained CNN model for visual feature extraction and appending multiple network layers for image classification has proven able to get better results compared to constructing the model from scratch. Nevertheless, in this work, automated network construction performs much better than employing classical ML techniques to learn the high-level representations of the deep features. Results show increments of 12.2% and 5.3% respectively for weighted and unweighted average F1 scores for a new problem domain by adopting automated EA and considering the similarities between the solutions in the proposed work for the Disaster dataset. Compared to the pre-trained model using Network Camera 10K dataset, the proposed method improved the weighted average F1 score by 3.3%. Seeing there is a significant improvement in F1 score for both datasets, we can conclude that the proposed framework selected and built the network model which could recognize more instances correctly for each class. The table also shows the final configuration of the hyperparameters that achieved the scores as demonstrated.

4.6.4 Conclusions

In this work, we aim to leverage the deep features from pre-trained CNN models in different applications without spending most of the effort in examining the characteristics of each task. A generalized framework is proposed to accommodate all datasets. By integrating EA and other techniques to support the automated searching process, the pro-

posed work determines the hyperparameters of a new neural network for one specific task after the best individual is selected. Overall, the experimental results have proven that a time-consuming task conducted by experts could be done by an automated process that surpasses human ability and reaches an optimal solution effectively. It should be noted however, that validation feedback needs to be provided lest the network select the best individual for an incorrect resultant task. Quadded neural nets could be applied to ameliorate this situation somewhat, but until the nets can be designed to reliably extract fundamental features and combinations of features, this possibility will persist. Again, the differential is attributed to "understanding" or the lack thereof. Higher-level evolution requires (self-referential) heuristics. An open question is how to represent and apply them in deep learning.

CHAPTER 5

MULTIMODAL DEEP REPRESENTATION LEARNING

Today, the World Wide Web including various media types such as video/image, audio, and text has increased a lot of attention in multimedia big data analytics [95, 114, 150, 187, 188]. Multimedia event detection and classification is a challenging task due to the amount and diversity of data required to be processed [41, 49, 189, 190]. Extracting discriminative features and integrating different sources of information are essential steps to achieve accurate multimedia classification [28, 191].

In this era, social media and web resources such as Facebook, Twitter, and YouTube generate a rapidly increasing amount of real-time information [106, 192]. Such user-generated data contains rich information that can provide a deep insight into what events are happening around the world. However, it has become more and more complex to be able to extract important information from such heterogeneous big data. One of the main challenges is how to leverage the various data modalities effectively.

When an important event such as a natural disaster happens, a large amount of video data are posted on the Web. For example, after the devastating hurricanes Harvey, Irma, and Maria happened, many users shared their videos and images on the Web and social media. Specifically, more than 155k videos were uploaded to YouTube about these three hurricanes. Hence, it is very difficult to search the videos most relevant to the user's interest. Multimedia big data techniques including multimodal data collection, analysis, and visualization have shown to be promising in handling and finding information from multimedia big data. They are also able to significantly enhance the development of the disaster management systems [122, 193, 194, 195, 196].

5.1 Multimodal Deep Representation Learning for Video Classification

This study performs a content analysis of disaster-related data provided on the Web, which can be further used for disaster information retrieval. For this purpose, a multimodal deep learning framework is proposed that utilizes different sources of information including visual, audio, and textual data. Unlike conventional fusion techniques (e.g., early fusion and late fusion), we propose a two-stage modality fusion approach which first analyzes the temporal information from both visual and audio data and then combines the textual information with the results from the first stage.

To the best of our knowledge, this is the first multimodal deep learning framework for natural disaster video analytics. Specifically, the contributions of this study are listed as follows.

- A new video-based dataset is collected for natural disaster management. The videos are collected from Youtube during hurricane Harvey. Along with the videos, the titles and descriptions of the videos are also collected to enrich the presentation of the dataset for multimodal analysis.
- A new multimodal deep learning framework is presented that automatically generates deep features from each modality using the most advanced deep learning models.
- A two-stage fusion approach is proposed, where a Convolutional Neural Network (CNN) is first applied to combine the temporal features and then the late score fusion generates the final classification.

Figure 5.1 illustrates the overall structure of the proposed framework which includes two stages of the fusion process. Different modalities and their corresponding deep learn-

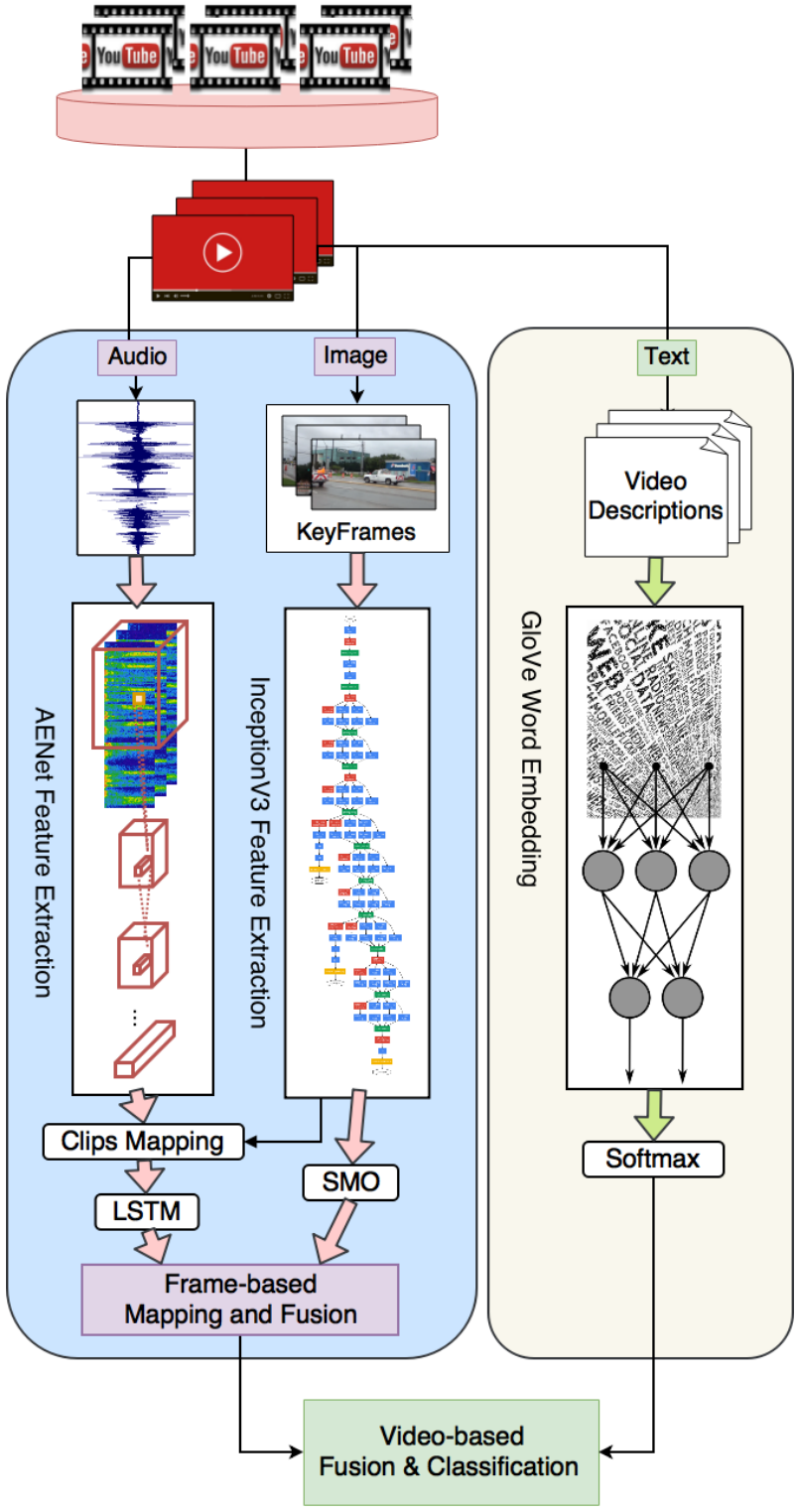


Figure 5.1: The proposed video classification framework including two-stage fusion for frame-level and video-level data integration. Frame-level audio and image features are fused in the first stage and later combined with video-level textual data in the second stage.

ing outputs are effectively integrated into this fusion process which happens at the frame level and video level. Video titles and descriptions come with the individual videos, so the text model performs the classification tasks with word embeddings on the video level. In other words, the textual model is trained and tested for the video level concepts. Comparing to the frame level information (e.g., keyframe images and audio clips), the video level description plays an important role in summarizing and making a conclusion of the complicated concepts into a more comprehensive concept. In this framework, the frame level classification focuses on the more specific concepts which later can be used for the video level classification. Most of the frame level concepts do not overlap with the highly summarized video level concepts since a single frame only includes a group of objects or simple scenes which cannot tell the main idea of the entire video. This proposed framework contains several main steps including preprocessing phase, training phase, and testing phase. The detailed descriptions of each step with the sub-model inputs and outputs are explained in the following sections.

5.1.1 Frame-based Image Model

Support Vector Machines (SVMs) are popular supervised learning methods for classification. Either softmax or SVM with a linear kernel is usually used as the prediction layer of the deep learning models. Linear SVM is less prone to overfitting than the non-linear kernel and useful in high dimensional spaces with a less number of samples. Sequential Minimal Optimization (SMO) is an advanced version of SVMs with faster algorithm training. The memory requirement for SMO is linear to the training set size, which allows SMO to handle extensive training sets. SMO avoids matrix computation and scales the training set size for various test problems. Therefore, SMO computation time is significantly less than the original SVM. On real-world sparse datasets, SMO can be more than

1000 times faster than the SVMs. Hence, SMO is the fastest for linear classification and for working with sparse datasets. In our proposed framework, a linear kernel SMO is used as the classification model on top of the pre-trained InceptionV3 features.

5.1.2 Frame-based Audio Model

The frame-based audio model maps the audio features extracted by the AENet, to the frame-based concepts. This model plays a role as a complement to the image model, which provides the distinguishing characteristics of the keyframes and thus enhances the overall performance of the proposed framework. For example, the concept "storm" is hard to be identified by images since it is invisible. In the proposed framework, the LSTM layer [197] is applied to analyze the temporal information of the sequences of the audio features mapped to each keyframe and bridge the gap between audio features and frame-based concepts. LSTM is an RNN, which is suitable to handle temporal data and generate classification from the model. Compared with conventional RNN, it has great advantages in avoiding gradient exploration and vanishing during the training.In the model, the feature sequences are first fed to three LSTMs, and then a fully connected layer attached with softmax activation is used to generate the result. Each LSTM layer includes 32 units. RMSprop optimizer is applied to train the model with the customized learning rate at 0.0001.

5.1.3 Multimodality Feature Mapping

For each audio wave file that we generate from the corresponding video, a sequence of audio features are produced by a set of clips with the same time duration. The input patch size that represents the duration is set to 2000ms, which is equal to the length of 200 frames as one frame represents 10ms. The output feature is a 1024-dimension matrix, and

the number of rows in the output matrix depends on the length of the original audio. Each row of the feature output contains half of the audio signal overlapped with the previous input, which is set by a sliding parameter in order to keep the continuous information. The window sliding step shifts 50% of the input patch size for the sequence generation. A simple illustration of the frame mapping is shown in Figure 5.2, and the details are presented in Algorithm 4.

Since only keyframes are selected to represent the frame-level concepts, we considered not using the complete set of audio features for the entire video to build the audio model. Therefore, based on the frame index, audio features can be mapped to the keyframes that we used for the image model. This mapping strategy provides the possibility of fusing the scores we get from the separate training models. The overall idea of the mapping function is to locate the keyframes using the frame timestamp along with the audio feature clips. The available frame range for each audio feature clip is defined by the lower_idx and upper_idx in lines 2 and 3, which are calculated using the pre-set audio feature sequence length (2δ), the pre-set shifting window length ($\mathrm{patch\_size}$), and the stride settings of the audio feature extraction phase (stride). In the calculation, the offset refers to the timestamp of the keyframe. If the offset is too small and thus the lower_idx is below 0, the sequence of features is padded by 0. Since the audio features are extracted using the entire video, so it can be used as the base map to allocate the positions of the stored keyframes. Furthermore, the audio clip in the training dataset will be assigned with the identical frame-based label if the keyframe exists.

5.1.4 Video-based Text Model

We built a text classification training model by using a pre-trained word embedding layer with a 1D convolutional neural network. Specifically, we choose the 200-dimensional

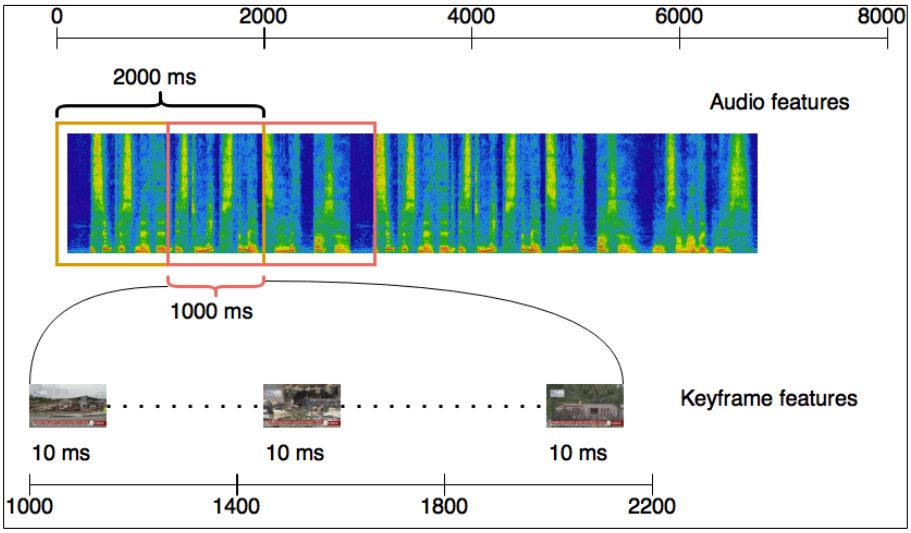


Figure 5.2: An example of frame-based feature mapping that shows how the window shifts with 100 frames (10ms/frame) to extract the formatted audio features. Also, the keyframes are mapped with the audio feature clips with the same frame scaler.

Algorithm 4: Frame-level Feature Mapping

for each keyframe in keyframe_list **do**
    lower_idx $\longleftarrow \left\lfloor \frac{\text{offset}-\text{patch\_size}}{\text{stride}} \right\rfloor + 1 - \delta$;
    upper_idx $\longleftarrow \left\lfloor \frac{\text{offset}}{\text{stride}} \right\rfloor + \delta$;
    feature_seq $\longleftarrow$ list();
    if lower_idx < 0 **then**
        for $i \in$ [lower_idx, -1] **do**
            feature_seq.append(0)
        lower_idx = 0
    for $i \in$[lower_idx, upper_idx] **do**
        feature_seq.append(features$_i$)
return feature_seq

GloVe embeddings of 400k words computed on a 2014 dump of English Wikipedia. First, all text samples in the dataset are converted into sequences of word indices which are merely integer IDs for the words. We consider the top 5000 most commonly occurring words in the dataset and truncate the sequences to a maximum length of 1000 words. The embedding matrix contains the embedding vector of each word index. The embedding matrix is loaded into a Keras embedding layer and set to frozen, which means it will not be updated during the training.

The model is further built on top of a 1D convolutional neural network, ending in a softmax output over our nine video categories. Some regularization mechanisms, like dropout layers, are used after both the first and the last layers to reduce the effect of overfitting. Adam optimizer is used to compile the model with a customized learning rate 0.0001. The kernel size for each layer is set to 5, which means we consider five contiguous words, which is ideally enough to represent the meaning of the phrase.

5.1.5 Frame-based Joint Representation

Figure 5.3 shows the framework generating the frame-based joint representation, where the frame-level results from both visual and audio modalities are mapped to the video-level concepts. Since both visual and audio classifications are performed on keyframes, the results generated for the same frame are highly related, and thus they are grouped together during the fusion. As shown in the left part of Figure 5.3, each visual score is placed between two audio features and each audio feature is placed between two visual features; while the order of both types of the features is kept. Correspondingly, the first convolutional layer configures the strides as 2 and the kernel size as 10. So, the features of the same keyframes are grouped and regarded as a union data representation. Moreover, since the visual and audio scores of the same keyframe are placed together, the stride 2

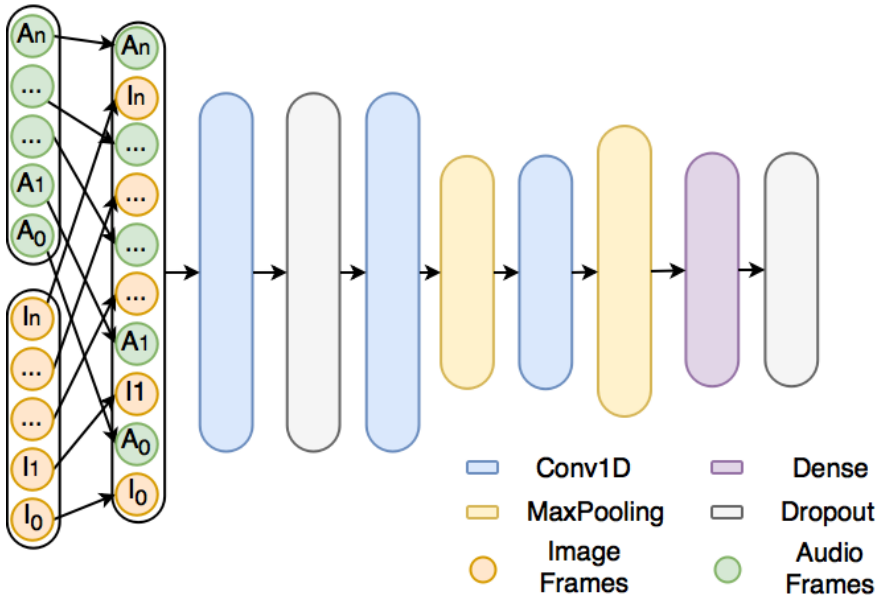


Figure 5.3: The framework of the proposed joint fusion method that combines the temporal features of the visual and audio data for keyframes and generates the video-level scores. The left part concatenates both data with re-ordering the features and the right part applies a CNN to analyze the temporal information.

and even the kernel size can ensure that the scores from different keyframes are treated equivalently. Along with the first convolutional layer, another two convolutional layers and the output layer with the softmax activation function are added to the network for temporal information analysis, where the joint representation is generated. The convolutional layers describe the relations between successive keyframes and apply proper weights to combine them and generate the video-level classification scores. This neural network is trained by Adam optimizer with the customized learning rate 0.0005.

5.1.6 Dataset Description and Preprocessing

Dataset

We create a new dataset which includes several complex semantic concepts related to disaster events. The complexity of the dataset is high since it includes not only the classic disaster scenes but also the activity concepts like emergency response, human reactions, as well as environmental surroundings [198]. Any video has a potential semantic link with the given disaster event included in this dataset, while accurately categorizing the videos provides more valuable information to the possible audience. For examples, when a hurricane warning is issued, people keep an eye on the latest situation change along with the storm track predictions. When the storm has passed by, people's attention changes to look for the damage situation and rescue activities around the affected areas.

Our dataset includes almost 400 Hurricane Harvey related videos with the corresponding text information. We started crawling the videos and their descriptions from Youtube when Harvey became the major hurricane and attracted a lot of public attention. The crawler kept gathering related videos for about one week until another significant storm Irma approached south Florida. This is common in all social media that when one event becomes the focus of attention, the previous one will lose the attention very fast. Thus, it is challenging to gather the time-sensitive event information afterward, and it requires a prompt response to the event. For each video we get from the keyword searching list on Youtube, we also use the recommendation functionality of YouTube to obtain the top 5 related videos to the crawler list to reduce the time of looking through the video list one by one. Together with the video, text information such as video descriptions was also gathered and stored in the XML format with different tags (e.g., title and description).

Table 5.1 shows the statistics information of the dataset in which both frame-level labels and video-level labels are provided. Generally, if the video contains a large num-

Table 5.1: The statistics of the dataset including frame-level and video level concepts

Concepts	# of images	Concepts	# of videos
Building Collapse	770	**Situation Reporting**	123
Flood	8354	**Emergency Response**	21
Human Relief	1427	**Human Relief**	44
Damage	1283	**Preparation**	12
Speak/Interview	2246	**Disaster Scene**	76
Preparation	468	**Demonstration**	32
Briefings	851	**Victim Situation**	25
Demonstration	807	**Damage Situation**	31
Emergency Response	1619	**Volunteer Activity**	25
Volunteer Activity	283		
Storm	2218		
Road Debris	220		
Regular Surrounding	2749		
Victim/Refugee	1793		
Daily Necessaries	453		
Animals	164		
total #	25705	**total #**	389

ber of keyframes, its frame-level concepts have more variety than the shorter one. For instance, the most complicated video in this dataset includes 12 classes out of the total 16 labels. However, there are still nearly 1/4 of the videos containing a single concept for all the keyframes. The uncertainty increases while the involved number of concepts also increases in a video. Therefore, it is a challenging task to summarize and perform reasoning about the storyline of the videos.

The video dataset is randomly separated into 20% testing and 80% training based on the video classes. At the same time, by considering the training and evaluation process for the image level models, the distribution of the image level classes is checked to ensure the average separation rate is also 20%. In that case, the entire group of keyframes belong to the training/testing video will be included in the training/testing image dataset. Table 5.2 summarizes the statistics of the training and testing sets as well as the P/N ratio for each concept.

Table 5.2: The statistics of the training and testing sets as well as the corresponding testing P/N ratio

Concepts	Train	Test	P/N test
Building Collapse	664	106	0.021
Flood	7099	1255	0.319
Human Relief	1092	335	0.069
Damage	1080	203	0.041
Speak/Interview	1972	274	0.056
Preparation	372	96	0.019
Briefings	702	149	0.030
Demonstration	581	226	0.046
Emergency Response	1401	218	0.044
Volunteer Activity	225	58	0.011
Storm	1401	817	0.187
Road Debris	191	29	0.006
Regular Surrounding	1814	935	0.220
Victim/Refugee	1420	373	0.078
Daily Necessaries	371	82	0.016
Animals	135	29	0.006

Several procedures are applied to clean the raw data. First of all, the related metadata is checked to see how the video is related to the targeted hurricane event. For example, the hurricane event happens only one or two weeks within a given time interval. Therefore, if the creation date of the video is older than the start time of that event, it is highly possible that the video is not relevant and should be removed from the dataset.

5.2 Sequential Deep Learning for Disaster-Related Video Classification

Each data modality has its own strengths and associated deep learning approaches and techniques. For audio data, models that are able to extract or predict natural sounds are very scarce due to the focuses on speech recognition and music classification. As for textual data, word embedding models show significant improvements both as feature

learning and language modeling techniques by representing the similarity between words and meaning through their closeness in the real-number vector space.

More recently, multimodal deep learning techniques [113] have been introduced to enhance the performance of deep models that focus solely on a single-modal data type. In this paper, we propose a multimodal deep learning framework that incorporates sequential information from audio and textual models, where different deep features are extracted from each modality using the pre-trained Convolutional Neural Network (CNN) models and word-to-vector technique. Subsequently, Long Short-Term Memory (LSTM) neural networks are applied to leverage the sequential relationships, particularly for text and audio data. We use CNNs to build our fusion model to incorporate the classification ranking scores produced by data from single modalities. Finally, the proposed framework is applied to classify the videos in a disaster-related video dataset as a certain disaster-related concept (flood, storm, etc.).

The contributions of the proposed framework are as follows.

- A multimodal deep learning framework that incorporates sequential information from both audio and textual models;
- For the audio model, an effective and efficient deep learning model is utilized to extract the most discriminative and high-level feature representations that we extend through a time distributed fully connected layer and the subsequent LSTM layers. For the textual model, a pre-trained word embedding layer is used with a stacked LSTM model to generate the video-level concepts; and
- A novel two-stage fusion technique is proposed based on the frame-level image, audio, and video-level information by building a CNN model. Most notably, the image model predictions are incorporated into the audio model to adjust the clas-

sification ranking scores based on the reliability of the different predicted sound classes.

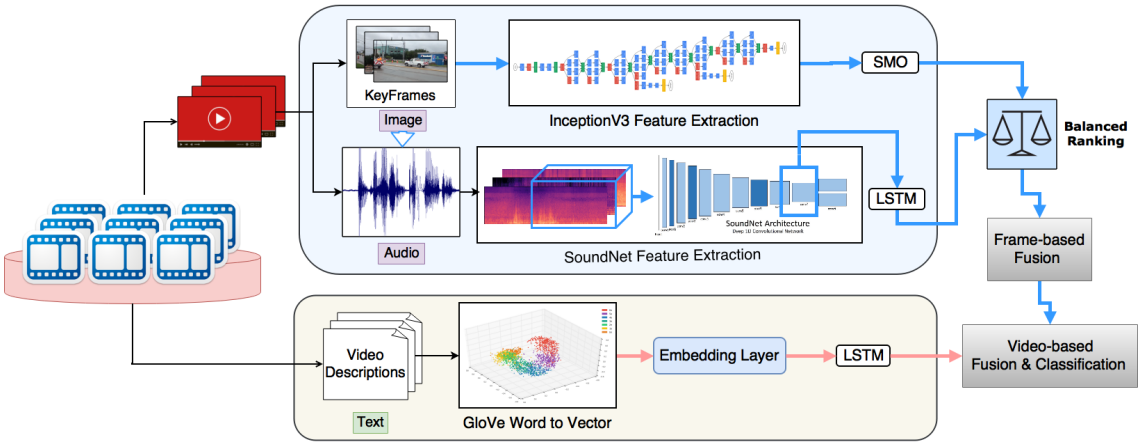


Figure 5.4: The proposed framework handling sequential information

Figure 5.4 illustrates the overall framework which includes three deep learning models for the different data modalities and a two-stage fusion model. The outputs from the corresponding deep learning models predict potential semantic concepts by providing ranking scores. The scores (or probabilities) are taken as the inputs by the fusion model, which considers both frame-level and video-level concepts.

5.2.1 Audio Clip Mapping

In order to use the labels (semantic concepts) of the key frames (images) as the references to the audio model, we first extract the full audio tracks from the raw videos with a sampling frequency of 16000 Hz. The metadata of the video and the frame numbers of the key frames are used to calculate the starting and ending points of the audio clips. Algorithm 5 shows the mechanism we used to slice the audio clips from the full audio tracks, which also guarantees that such clips are generated accurately in order to get sound waves that really match with the visual concepts. The inputs include a list of raw

Algorithm 5: Get audio clips from full audio tracks

```
1  begin
2      audio_metadata ⟵ initialize();
3      foreach track ∈ audio_metadata do
4          track_d ⟵ load_audio(track.vid);
5          fps ⟵ get(video_fps, track.vid);
6          ref_time ⟵ track.fid/fps;
7          if ref_time − span < 0 then
8              save_clip(track_d, 0, duration, track);
9              return;
10         else
11             start ⟵ ref_time − span;
12         track_length ⟵ len(track_data/1000);
13         if ref_time + span > track_length then
14             last ⟵ track_length − ref_time;
15             start ⟵ ref_time − (duration − last);
16             end ⟵ track_length;
17             save_clip(track_data, start, end, track);
18             return;
19         else
20             end ⟵ ref_time + span;
21         save_clip(track_d, start, end, track);
```

audios with the corresponding *frames per second (fps)* rate (frame rate) for the associated videos.

The *initialize()* function in line 2 generates a key-value paired dictionary *audio_metadata*, where the keys represent the video ID (*vid*) and the values represent the frame ID (*fid*) for that specific video. The variable *duration* holds the value in seconds of the desired audio clip. Additionally, *span* holds the duration of each interval before and after the specified key frame. Starting from line 3, the program loops to process the entire dictionary. The first steps for every audio track (*track*) consist of: a) loading the audio through Pydub (line 4); b) getting the frame rate of the video (line 5); and c) calculating the temporal reference for that specific *fid* (line 6). The variable *ref_time* contains the value in seconds of the current key frame. Lines 7-9 handle key frames that are close to the beginning of

the audio track. Similarly, lines 13-18 handle case where the key frame is close to the end of the audio track. This guarantees that all the audio clips have a duration of exactly eight seconds. In line 14, the variable *last* holds the value in seconds from the current key frame being processed until the end of the audio track. We use this value to determine how much we need in order to build an eight-second audio clip. *save_clip* is a helper function that slices and saves the audio clip through Pydub [1].

5.2.2 Frame-based Model

Linear kernel Support Vector Machines (SVMs) are popularly used to replace the softmax layer as the prediction layer of several deep learning models. As an advanced version of SVMs, the Sequential Minimal Optimization (SMO) algorithm speeds up the training process by avoiding matrix computations and scaling the training set size for different test problems. Our proposed framework uses an SMO classifier with linear kernel to process the deep features for the key frames that we obtained through InceptionV3 and outputs the label prediction probabilities for each instance. We use these probabilities as the input to the first stage of our fusion model and at the same time, as a guide for the model to take the scores from the audio model with an adjustable reliability.

In order to overcome the limited capability of the existing audio models, which detect few concepts as compared to the image models, we extended the SoundNet model to detect natural sounds and other activities by considering the sequential characteristics of the features. The audio model presents a higher accuracy in comparison to using SMO as the output layer on all the frame-level concepts.

[1] Available at https://github.com/jiaaro/pydub

The audio model aims to improve the performance of the frame-level classification by adding the capability of detecting scenes that are easily recognizable through sounds but harder to recognize through vision. The audio model consists of a fully connected Dense layer with 512 outputs, wrapped in a TimeDistributed layer, and then the subsequent LSTM and output layers. The Dense layer on top of an LSTM performs input compression before running it through the subsequent RNNs. Based on the output we generate from the audio features extraction step, the LSTM model takes the input as 5 timestamps, with each one containing 512 features. The LSTM model takes sequential data to learn how the changes of features in a temporal manner generate a better understanding of the audios in order to classify different kinds of sounds. RMSprop optimizer is used to compile the model with a customized learning rate of 0.0001.

5.2.3 Balanced Cross-Modal Ranking

The fusion model has two stages. The first stage concludes the frame-level concept predictions (for both image and audio models) and outputs the probabilities for potential video-level concepts. The latter one takes the textual model outputs that predict the video concept directly and integrate them with the results from the previous fusion stage to generate the final output.

Each single-modality has its own limitations. For example, the image model has the advantages on detecting static objects, but presents a limitation in scene detection which requires temporal information. On the other hand, the audio model has the advantages on detecting natural and human sounds, but the complexities arise when trying to detect activities that only produce few sounds or noise, such as near-silent situations. Inspired by the strengths and advantages of the different models, we propose a fusion algorithm that considers both the reliability and limitation factors of each modality for different cases.

The purpose is to balance the ranking scores which might dominate the potential concepts in different situations.

Algorithm 6 depicts how the ranking scores of the audio model are changed based on the predictions from the image model before the fusion stage. For each key frame f in video F, we examine the ranking scores and get the predicted labels from both visual and audio (A) models ($\mathcal{L}_1$ and $\mathcal{L}_0$, respectively). The most unlikely concept to be detected, which is identified by the lowest score among all concepts C, is also saved in variable m. If the predicted labels match with each other, the audio model prediction is trusted and a dominating concept penalty factor is applied to the scores, as shown in Equation (5.1). If a mismatched prediction is detected, the scores from the audio model need to be determined, considering the limitations from both models. $B_{f,c}$ represents the balanced ranking score for the frame-based concept c ($c \in C$) and the associated key frame f, where $\mathcal{L}_0$ is the predicted concept from the audio model ($A_{f,c}$) and $\mathcal{L}_1$ is the predicted concept from the image model ($I_{f,c}$). $|\cdot|$ represents the cardinality of the set.

$$B_{f,c} = \begin{cases} \frac{A_{f,\beta(c,\mathcal{L}_0)}}{Rank(c)} & \text{for } \mathcal{L}_0 = \mathcal{L}_1 \\ \frac{A_{f,c}}{Rank\prime(c)} & \text{for } c = \mathcal{L}_0 \neq \mathcal{L}_1 \\ \frac{A_{f,c}}{|C|} & otherwise \end{cases} \tag{5.1}$$

$$\text{where } \beta(c, \mathcal{L}_0) = \begin{cases} c & c = \mathcal{L}_0 \\ m & otherwise \end{cases}$$

$$Rank(c) = \begin{cases} 1 & \text{for } c \in \{\text{OtherSounds}\} \\ ln(|R(c)|) & otherwise \end{cases} \tag{5.2}$$

$R(c) = \{\{c \in \text{NaturalSounds}\} \cup \{c \in \text{HumanSounds}\}\}$,

$$Rank\prime(c) = \begin{cases} |\overline{N \cup H}| & \text{for } c \in \{N\text{: NaturalSounds}\} \\ |H| & \text{for } c \in \{H\text{: HumanSounds}\} \\ |\overline{O \cup H}| & \text{for } c \in \{O\text{: OtherSounds}\}. \end{cases} \tag{5.3}$$

$\beta(c, \mathcal{L}_0)$ is an activation function which selects the concept with either the highest or the lowest ranking score from the audio model, depending on whether the concept is a match or not. The function $Rank(c)$ returns a penalty factor for the frame-level concepts that belong to either human sounds or natural sounds (as shown in Equation (5.2)). Natural logarithm is used in the equation in order to guarantee the return of a penalty factor by preventing the divisor in the first case of Equation (5.1) from being equal to 1. Considering the capability of the audio model, no penalty factor will be applied when there is a match for other sounds. Equation (5.3) is the $Rank\prime(c)$ function that returns a coefficient based on the number of image concepts associated to the current audio concept in case of a concept mismatch.

Based on our observations, the audio model can better differentiate human and natural sounds from other sounds. However, the ranking scores might be dominated by natural sounds since they are frequently present as the background sound in most of the disaster-related events and activities. To remediate this imbalance scenario, the penalty factor for different sound types will be the opposite of the accuracy of the model. This way, natural sounds will always take the biggest penalty factor compared to other sounds.

The grouped and balanced ranking scores from both image and audio models are the inputs of the first stage of the fusion model, with the first convolutional layer configured with a stride of 2 and a kernel of size 10. The network is trained using a RMSprop optimizer with the default learning rate to preserve the sequential capabilities of the data. During the second fusion stage, the text classification model results will be integrated with the predicted conclusions from the frame-level modalities. Since there are some videos

Algorithm 6: Frame-level audio rank balancing

for $f \in F$ **do**
  $\mathcal{L}_0 \longleftarrow \underset{c \in C}{\operatorname{argmax}} A_{f,c}$;
  $\mathcal{L}_1 \longleftarrow \underset{c \in C}{\operatorname{argmax}} I_{f,c}$;
  $m \longleftarrow \underset{c \in C}{\operatorname{argmin}} A_{f,c}$;
  for $c \in C$ **do**
    if $\mathcal{L}_0 \neq \mathcal{L}_1 \bigcap c = \mathcal{L}_0$ **then**
      $A_{f,c} = min(A_{f,m}, B_{f,c})$
    else
      $A_{f,c} = B_{f,c}$

that do not provide text information, our proposed framework also has the ability to deal with missing values at this stage. If the text information for the related video exists, the prediction from the text model will be integrated into the network. Otherwise, the results from the previous model's outputs will be directly used. In this model, two Dropout layers (with 0.7 and 0.4 dropout rates, respectively) are added after the Flattened and Dense layers in order to prevent overfitting. The network is also trained using a RMSprop optimizer with default learning rate.

5.2.4 Experiments and Analysis

The experiments were conducted by using a dataset which includes almost 400 Hurricane Harvey-related videos with associated text information, namely video title and description, which we collected from YouTube in 2017. Table 5.3 shows the frame-level, video-level concepts, and general audio concepts (grouped and mapped to the frame-level concepts). The dataset is split into training (80%) and testing (20%) sets randomly on the condition of maintaining the frame-level concepts within an approximately similar distribution. At the same time, the key frames (images) and audio clips that correspond to a video will only appear in either the training or testing set.

Table 5.3: Image (frame-level), general audio, and video-level concepts across different modality datasets.

No.	Video-shot Keyframe Concepts	Video Concepts
1	Building Collapse	Situation Reporting
2	Flood	Emergency Response
3	Human Relief	Human Relief
4	Damage	Preparation
5	Speak/Interview	Disaster Scene
6	Prepare	Demonstration
7	Briefings	Victim Situation
8	Demonstration	Damage Situation
9	Emergency Response	Volunteer Activity
10	Volunteer Activity	
11	Storm	
12	Road Debris	**Video-shot Audio Concepts**
13	Regular Surrounding	Natural Sounds (**2, 11**)
14	Victim/Refugee	Human Sounds (**5, 7, 8**)
15	Daily Necessaries	Other Sounds
16	Animals	(**1, 3, 4, 6, 9, 10, 12-16**)

The evaluation metrics used in our experiments of multi-class classification are Accuracy (ACC.) and Label Ranking Average Precision (LRAP) [172]. LRAP was originally used in multi-label ranking problems, where the goal is to give better ranks to the labels associated to each sample. In this study, there is exactly one relevant label per sample, which makes LRAP equivalent to the mean reciprocal rank. Let I be the total number of instances and $|C|$ be the total number of concepts. Formally, given a binary indicator matrix of the ground truth labels $y \in \mathcal{R}^{I \times |C|}$ and the score associated with each label $\hat{f} \in \mathcal{R}^{I \times |C|}$, the label ranking average precision is defined as:

$$LRAP(y, \hat{f}) = \frac{1}{I} \sum_{i=0}^{I-1} \sum_{j:y_{ij}=1} \frac{|\mathcal{L}_{ij}|}{\text{rank}_{ij}} \tag{5.4}$$

with $\mathcal{L}_{ij} = \left\{ k : y_{ik} = 1, \hat{f}_{ik} \geq \hat{f}_{ij} \right\}$, $\text{rank}_{ij} = \left| \left\{ k : \hat{f}_{ik} \geq \hat{f}_{ij} \right\} \right|$.

As shown in Table 5.4, the prediction using image features alone achieves higher ACC. and LRAP compared to the prediction using audio features. Through the LSTM

Table 5.4: Evaluation results of balanced-ranking fusion compare with single modality and simple fusion models' performance

Methods	ACC.	LRAP	# of concepts
Frame-based audio (SMO)	0.261	0.448	16
Frame-based audio (LSTM Model)	0.283	0.470	16
Image Model	0.346	0.534	9
Audio+Image Fusion	0.345	0.525	9
Video-based text (LSTM Model)	0.366	0.530	9
Balanced-Ranking Fusion	**0.444**	**0.596**	9

audio model, we show the strength of our sub-model compared to the simple output layer (SMO) that does not consider sequential information. However, if we use our sequential fusion model directly on the image and audio models' outputs, it shows how the contradiction of the predictions in different modalities will degrade the results of the entire framework, which leads to a decrease in accuracy. By applying our proposed two-stage fusion model, the fusion results gain strength from both image and audio models and reduce the effects of contradicting predictions. The proposed framework, through the fusion of predictions from three modalities, improves the accuracy by more than 10%.

5.2.5 Conclusions

Multimodal deep learning has recently attracted a lot of attention. This paper proposes a novel multimodal deep learning framework that considers sequential information from both audio and textual models. Furthermore, a two-stage fusion technique is proposed that utilizes the frame-level image, audio, and video-level information by building a CNN model. In our experiments, we demonstrate how the proposed framework improves the accuracy from single-modal models and illustrate the capability of fusion strategies by

taking into account the prediction contradictions across modalities in order to balance the reliability for different class predictions.

CHAPTER 6

MULTIMODAL FUSION FOR SEMANTIC CONCEPT DETECTION

Vast amounts of multimodal data (image, video, and text) are generated on a daily basis by users through personal devices and social networking services. Classifying massive amounts of single-modal data is an ongoing research field that has gained benefits from the advances in computer vision, audio classification [147], text recognition [199], and natural language processing [101]. However, as the magnitude and capabilities of data generation and collection grow exponentially, more reliable and cutting-edge classification methods are needed in order to reap the benefits of the knowledge that can be attained, from not only each single modality alone but also multiple modalities. Additionally, the new challenges that surface when trying to acquire the useful information from multimodal data demand improved techniques to obtain more accurate classification results.

6.1 Decision Fusion

To address such challenges, a Feature Affinity based Multiple Correspondence Analysis and Decision Fusion (FA-MCADF) framework is proposed. The important relationships among features within each feature group are preserved without being affected or counteracted by other representations of features that are less correlated globally. In the decision fusion stage, the classification results from several FA-MCA classifiers are fused. The criterion that decides the best threshold in the testing phase is used here to evaluate the reliability of each training model. The F1 scores (s_f) [159] calculated during the training phases are optimized by the best threshold selection. The average F1 score ($\bar{s}$) and the standard deviation (std_F) accumulated by the F feature groups decide whether the specific group of features is a good representative of the concept. The Bessel's correction

[200] is applied to the standard deviation calculation as shown in Equation (6.1). F is also the number of the basic FA-MCA classifiers which correspond to the F feature groups.

$$std_F = \sqrt{\frac{1}{F-1}\sum_{f=1}^{F}(s_f - \bar{s})^2}. \tag{6.1}$$

In the proposed framework, each feature group is considered as an equally contributed input to the final decision. Meanwhile, the uncertainty of the contribution for each representation space makes the fusion scheme flexible, as illustrated in Equation (6.2) and Equation (6.3). Pn' is the final label prediction set for each testing data (te). If the instance is predicted as negative in any feature group, the prediction value will be added to the output set Pn' that takes the z-score value (γ_f) and the sum of the values of all the basic classifiers. For every testing instance, each FA-MCA classifier produces one prediction result. The prediction result is a binary value, either 1 representing negative (not belonging to the concept of interest) or 0 representing positive (belonging to the concept of interest). For example, if there are four basic classifiers that classify one instance as negative, the final score will be summed up to at least four, since a smaller absolute z-score value of a specific classifier represents a higher reliability.

In addition, since the 99.7% confidence interval is represented between the z-score values of -3 and 3 [201], the α value is set to 3.5 empirically to eliminate the effect of abnormal values and keep as much information as possible.

$$Pn'^{te} = \sum_{f=1}^{F}(Pn_f^{te} + \gamma_f) \tag{6.2}$$

$$\gamma_f = \begin{cases} \frac{Pn_f^{te}}{F+|zscore_f|} & |zscore_f| \leq \alpha \\ Pn_f^{te} & otherwise \end{cases} \tag{6.3}$$

The decision fusion scheme mainly focuses on better predicting the negative instances, as we would like to keep as many positive instances as possible. Algorithm 7 illustrates

the idea of how to utilize the prediction results based on the normal distribution among all basic classifiers. The prediction set Pn is a $Te \times F$ matrix which includes F prediction results for Te testing instances. The z-scores of the basic classifiers are calculated in line 4 to decide the reliabilities.

Algorithm 7: Decision Fusion Scheme

1 **DFcal** $(Pn, s, \bar{s}, std_F)$
  inputs : The negative label prediction set Pn of each feature group F $\{Pn_f^{te} | f = 1, \cdots, F; te = 1, \cdots, Te\}$; the training set F1 score set s= $\{s_f | f = 1, \cdots, F\}$; the average F1 score $\bar{s}$; and the F1 scores' standard deviation std_F.
  output: The combined negative label prediction set Pn' $\{Pn'^{te} | te = 1, \cdots, Te\}$
2 **forall** Pn_f^{te} *(*$f = 1, \cdots, F$*)* **do**
3 //Calculate each F1 score's z-score;
4 $zscore_f = (s_f - \bar{s})/std_F$;
5 **forall** Pn_f^{te} *(*$te = 1, \cdots, Te$*)* **do**
6 Calculate Pn'^{te} using Equations (6.2) and (6.3); ;
7 Calculate β using Equation (6.4);
8 **forall** Pn'^{te} *(*$te = 1, \cdots, Te$*)* **do**
9 **if** $Pn'^{te} \geq \beta$ **then**
10 Pn'^{te} is negative;
11 **else**
12 Pn'^{te} is positive;
13 **return** Pn'^{te};

After accumulating the prediction results in line 6, the final decisions are made though a threshold β calculation using Equation (6.4) in line 9. When $\bar{s}$ is close to 1, the final classification result is considered to be trustable with a lower accumulated value. By setting the z-score value (z) to -2 in Equation (6.4), the mean value is shifted to the left with 2 standard deviation values. That is the smallest value between 0 and 1 in the 95% confidence interval and is considered as a fault tolerance number. This number is applied to half of the classifiers in order to decide the threshold of the summation, which indicates the negative label. Namely, at least half of the classifiers should classify an instance as

		Flood	Human Relief	Damage	Training Program	Diasater Recovery	Speak	Interview	Average
FA-MCA	**Pre**	70.25%	34.18%	71.91%	68.61%	70.32%	82.91%	68.79%	66.71%
	Rec	61.67%	30.52%	72.92%	65.48%	68.90%	97.78%	48.95%	63.74%
	F1	46.62%	24.45%	64.80%	45.07%	52.69%	87.42%	29.54%	50.08%
FA-MCADF	**Pre**	45.26%	34.53%	71.77%	68.53%	70.30%	82.93%	68.82%	63.16%
	Rec	85.84%	49.29%	96.06%	61.11%	97.48%	99.85%	78.63%	81.18%
	F1	**50.90%**	**28.08%**	**73.49%**	42.43%	**71.91%**	**88.45%**	**57.98%**	**59.03%**

Table 6.1: FA-MCADF performance compare with using single FA-MCA classifier

negative with a better z-score when the instance is a negative instance.

$$\beta = F - \frac{(\bar{s} + z * std_F)}{F} * \frac{F}{2} = F - \frac{(\bar{s} + z * std_F)}{2} \tag{6.4}$$

6.1.1 Experimental Results

In the last three rows of Table 6.1, the FA-MCADF framework is used to boost the performance by separating more than 700 dimensions of features into four feature groups (illustrated in Figure 4.1) that run FA-MCA models independently. Each FA-MCA model handles approximately 1/4 of the features, thus speeding up the learning process. A decision fusion scheme (Algorithm 7) is proposed as the final step to generate the final classification decision. The FA-MCADF framework achieves the best in all the evaluation metrics in comparison to the other classifiers in the experiments. The average recall and F1-score values elevate 17.44% and 8.95% more in comparison to the single FA-MCA model. Figure 6.1 visualized the effect of applying decision fusion after FA-MCA on separated feature groups.

6.1.2 Conclusions

Disaster-related concept detection does not limit to disaster events. It also includes various concepts that are critical disaster information, such as disaster preparation training, disaster recovery, and disaster damage situation. Since the correlations between those

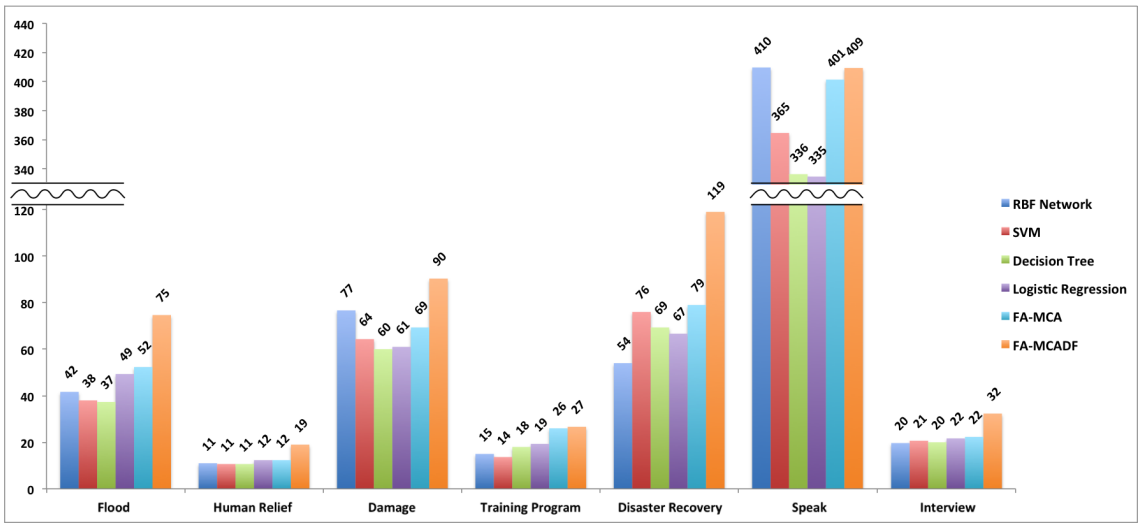


Figure 6.1: Number of True Positives obtained from each classifier

concepts are higher than diverse disaster events, it makes the classification task more challenging. To tackle this challenge, the FA-MCADF framework is proposed to consider the relationship between features within each feature group to eliminate the situation when some features dominate during the feature weight generation process. As a result, critical features are selected and weighted based on their ranks. The decision fusion scheme allows a scalable number of feature groups to run the classifiers separately, which reduces the negative effect among the features that belong to different representation levels. Comparing with the decision tree and SVM classifiers, the experimental results show significant improvements for all the evaluation criteria, which means that the proposed framework truly holds the importance of the features when detecting the interrelated concepts. However, there is still some improvements that can be further carried out.

In the future, this framework will be further extended and tested for more concept detection applications. Multi-modality features include high-level features, like audio, spatio-temporal and textual information, also can be included to improve the concept detection performance [202]. In addition, the latest cluster computing techniques (i.e.,

Apache Spark) can be included to build up a parallel framework to reduce the computation time, which is worth considering when processing large datasets [203, 204].

6.2 Multimodal Fusion

The fusion model is designed in two-stage to handle the frame-level and video-level multimodal representations. The first stage takes the frame-level classification results as the input and generates a joint representation for the visual and audio data, mapping the frame level classes to the video level classes. In the second fusion stage, namely, the video-level late fusion, textual results are combined with the audio-visual results from the previous stage to generate the final video classes.

Frame-level vs Video-level Concept Relationship

Based on the baseline tests we did (please refer to the experimental results in Section 6.2.1) by directly assigning the video label to all the keyframes, it does not generate a reasonable performance since every single frame can have different concepts. However, if some analysis is carried out on how the image labels should be grouped to conclude one video, the concepts can be associated together and a new label can be generated. For example, we have identified two frame-based concepts from the same video including flooding and one kind of human activities (e.g., emergency response, humanitarian relief, volunteer activities, etc.). Then, it is highly possible that the images classified as victims happen where the images of humanitarian relief exist. Hence, we can conclude that the video using the various concepts detected from its individual keyframes by following some domain knowledge rules.

In our proposed framework, we try to use a neural network to mimic the rule-based learning from the frame-based features and come up with the rules automatically without

applying the domain knowledge. To do so, there are some significant challenges. Since the model is going to be built based on the features, there is no specific high level semantic regulation involved to set the model. Furthermore, different concepts might have opposite dominate features, so our framework needs to handle it and makes the model robust regardless of the characteristics of the concepts.

Different video sources make the qualities of the videos vary. For instance, videos uploaded by normal users are usually short and contain fewer concepts, since they usually use their mobile devices to capture the surrounding environment. On the contrary, if the video is uploaded by some official sources, like CNN.com, it might contain more concepts to provide the more detailed information about the entire situation in different locations or from multiple official departments. The portion of various concepts in a single video also affects the final results.

Video-level Late Fusion

The frame-level concepts are helpful to bridge the semantic gap between raw video data and its concept. However, the real-world datasets have the imbalance problem that the distribution of the number of instances with different concepts has a long tail and few majority concepts take up the most portion of the dataset. This problem is much more severe in the frame-level classification since the frame-level majority concepts appear in the minority video concepts, but not in the opposite way. The objective function of minimizing the loss can lead to the bias towards the majority concepts which have a much higher accuracy value than the minority ones. Hence, the joint representation of the visual and audio data is more likely to have better results on the majority concepts while it may not perform well on minority ones as the textual model does. Therefore, in the final stage, the late fusion is performed based on the P/N ratio, the ratio of the number of positive to negative samples, of each video-level concept.

For each video v_i in the testing dataset V, each model generates the score for each concept to represent the likelihood that the concept appears in the video. We denote the score as $s_{x,i,j}$, where $x \in \{0, 1\}$ refers to the model type, $x = 0$ means that the scores come from the joint representation, $x = 1$ means that the scores are generated by the textual features, i refers to the video ID, and j refers to the concept ID. For example, $s_{0,1,2}$ is the score generated by the joint representation for video v_1 and concept c_2. Based on each single model, a concept with the highest score $c_{\mathcal{J}_x}$ is proposed for the final decision, where $\mathcal{J}_x = \underset{c_j \in C}{\operatorname{argmax}} s_{x,i,j}$. Let R_j be the P/N ratio of concept c_j and T be the threshold to determine whether it is the majority or not. That is, the concepts c_j with the P/N ratio $R_j \geq T$ are considered the majority ones. Based on the difference in performance, if both models propose the majority concepts, the proposal from the joint representation is preferred. If both propose the minority concepts, the proposal from the textual model is preferred. Otherwise, the likelihood values will be compared for the final decision. Since in YouTube, some of the videos do not have its textual description, we will utilize the joint representation results directly when this happens. The overall fusion approach is illustrated in Algorithm 8.

By using this proposed late fusion method, the final scores are determined by the proper model and the results can be combined by utilizing the advantages of both models. In addition, the missing value problem is handled using the proposed joint representation framework.

6.2.1 Evaluation Results

We conduct several experiments to show the effectiveness of the proposed framework compared to several existing methods. These experiments can be divided into two main techniques: single modal classification and regular fusions. The evaluation metrics used

Algorithm 8: Video-level late fusion

for $v_i \in V$ **do**
  $\mathcal{J}_0 \longleftarrow \underset{c_j \in C}{\operatorname{argmax}}\, s_{0,i,j}$;
  $\mathcal{J}_1 \longleftarrow \underset{c_j \in C}{\operatorname{argmax}}\, s_{1,i,j}$;
  $m \longleftarrow \underset{x \in \{0,1\}}{\operatorname{argmax}}\, s_{x,i,j}$;
  if v_i has textual description **then**
    if $R_{\mathcal{J}_0} \geq T \bigcap R_{\mathcal{J}_1} \geq T$ **then**
      return $c_{\mathcal{J}_0}$
    else if $R_{\mathcal{J}_0} < T \bigcap R_{\mathcal{J}_1} < T$ **then**
      return $c_{\mathcal{J}_1}$
    else
      return $c_{\mathcal{J}_m}$
  else
    return $c_{\mathcal{J}_0}$

in these experiments for multi-class classification are accuracy and mean Average Precision (mAP). The mAP is calculated as follows.

$$\text{mAP} = \frac{\sum_{q=1}^{N} AveP(q)}{N} \tag{6.5}$$

where N is the total number of classes, and $AveP(q)$ is the area under the precision-recall curve for a class q.

Table 6.2: Evaluation results of the proposed two-stage fusion framework

Methods	ACC.	mAP	# of classes
Frame-based image features	0.471	0.226	16
Frame-based audio features	0.226	0.070	16
Frame-based early fusion	0.430	0.208	16
Video-based image features	0.109	0.107	9
Video-based audio features	0.270	0.132	9
Video-based textual features	0.352	0.189	9
Video-based early fusion	0.167	0.118	9
Video-based joint representation	0.444	0.219	9
Proposed framework	**0.518**	**0.237**	9

First, the performance results of the frame-based image and audio features are generated to show how every single modality performs on this dataset. As shown in Table 6.2, the image features alone can achieve higher accuracy and mAP values than the audio features. This shows that the visual information plays a significant role in image/video classification. Thereafter, we combine these two features using early fusion and classify them using an SMO classifier. The performance result of this combination is shown in the third row of Table 6.2. However, this combination does not really improve the overall accuracy and mAP performance which illustrates why a better fusion method is needed for such complex multimodal datasets. In the next series of experiments, we compare video-based features including audio, image, and text with the proposed fusion model. As can be inferred from the table, all the performance results are significantly dropped when the video labels are used directly to train the model compared to the frame-level classification. For instance, the accuracy of the frame-based visual features is four times higher than the ones in the video level. Unlike the frame-level results, the model trained based on only the audio features performs better than the one with visual features in the video-level classification. This can be due to the large variations of image frames with different labels in a video which cannot be easily mapped to video-level classes while audio clips are usually very similar in the whole video. This phenomenon explains the necessity of mapping between image frame labels to video level classes and the multimodality integration. Regarding the textual features, it can relatively improve both accuracy and mAP values compared to other data modalities because video description and title are usually related to the final class. However, based on our experiments, there are still many unrelated descriptions or titles for videos, which reduces the overall video classification accuracy to 35%. Therefore, a fusion model can leverage all the modalities and enhance the detection performance. Two fusion models are used as the benchmarks, namely “early fusion” and “joint representation”. The former combines the keyframes audio and image

features before training the SMO classification; while the latter is a common approach to combine various data modalities inside the deep learning by concatenating data modalities representation. The results show how the joint representations can boost the final prediction results compared to a simple fusion approach. This motivates us to design a novel two-level fusion approach to leverage both frame-based image and audio features as well as video-based textual features. As a result, the proposed framework can enhance the detection performance compared to all other methods, which shows the effectiveness of this framework. Specifically, the final accuracy and mAP values reach to 0.518 and 0.237, respectively.

6.2.2 Conclusions

Deep learning has been widely applied to many real-world applications. Recently, it has been leveraged for multimodal data analysis using techniques such as early fusion or joint representation. Nonetheless, there are still challenges as how to effectively integrate multiple modalities using a general framework which learns the association between them. This paper proposes a novel framework that leverages the most advanced deep learning techniques to generate the features and data representation from various data modalities. This information is combined in two-stage using a new multimodal representation learning based on deep neural networks. This framework is particularly applied to a new dataset presented in this paper which includes natural disaster videos. The dataset contains three data modalities (visual, audio, and textual) and provides a real-world challenging scenario. In the experiment, it is shown how the proposed framework advances the state-of-the-art models including single modal deep learning and existing fusion models. In particular, it improves the classification accuracy performance more than 16% and 7% compared to the best results from single modality and conventional fusion approaches,

respectively. In the future, we intend to enhance the dataset by collecting more videos from natural disasters and increase the number of classes in both video level and image level.

6.3 A Video-aided Semantic Analytics System for Disaster Information Integration

6.3.1 Motivation

There have been many disasters in recent years. Both natural hazards and man made disasters cause huge damages on properties and human lives. When a disaster approaches, time becomes vital, since the emergency operation center (EOC) needs to update the situation reports immediately in order to provide the latest situation evaluation. The situation reports are mostly not too long and contain only highly abstracted critical information. Hence, many details are ignored, which in fact can be important for the emergency management personnel.

Multimedia data, including images and videos, can provide lots of useful information to aid the understanding of the situation reports due to its rich semantics. In particular, video data has become more and more popular since the rapid development of the Internet makes the transmission of large videos possible. Furthermore, video data has image frames, audio and motions, which better assist in the understanding and visualization of the text-based situation descriptions. In addition, the use of mobile devices can also be very helpful, since during the disaster event, people can capture and share the most current situation relevant data as fast as possible. This enables people to have a plan in an early stage instead of being trapped in the field.

Besides the potential enhancements from the front-end, there are many back-end optimizations (for the server-side) that can be considered. The enhanced web-based system could handle the video concept detection task by integrating with any machine-learning framework using standardized outputs at the back-end. It would enhance the system capability that provides as many related multimedia data as possible to expand the details that assist in the assessment of the current situation. Several automated functional triggers in the server-side can be implemented in order to further elevate the performance of the system. The triggers are responsible for different types of inputs. One would be to launch when new items are received, which aims to avoid losing any relationship with the existing stored items.

In this demo, we present a system that integrates situation reports and disaster-related multimedia data and provides an iPad application that conveys all the information via a unified and intuitive graphical interface. Moreover, the server-side of the system provides several interface components that integrate the video concept detection model and automatic event triggers to improve the performance of the semantic integration procedure.

6.3.2 System Architecture

The proposed system has a front-end client side that consists of an iPad application; the server-side contains a JSP-based API, and a database server. The JSP-based API mainly contributes to the indirect communication between the database and the client side by handling XML-based requests and information retrieval [205].

The database stores all the data related to the situation reports and the multimedia data as well as the user-account information. Its schema models the semantic relationship between the situation reports and the multimedia data. The situation reports cover one or more geographic *locations*, which are in turn depicted by videos taken at the disaster

area. The videos that were taken in one geographic *location* could contain several semantic *subjects*. For example, in the scenario of the storms affecting Alabama, geographic *locations* may be Tuscaloosa, Birmingham, and Hackleburg. Videos may describe the event or activity of one *location* before or after the natural hazard happened. The *subject* of a video depicting a location after the impact of the disaster could include the types of activities that were taken at the location. For example, the subjects of post-hazard videos may be "human relief", "disaster recovery", etc.

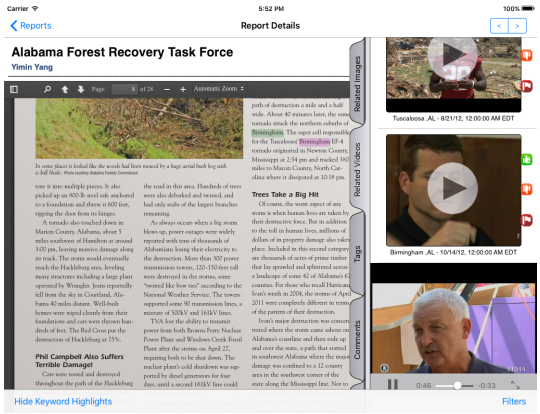
The presented system semantically associates the situation reports and related multimedia data through the location and subject entities. The geographic locations of the captured videos and report documents are extracted using the GATE framework [206]. In addition, a disaster-related video concept detection framework [207] as an independent component, is also integrated to the system, which helps to automate the entire process of semantic relationship deployment.

6.3.3 Demonstration

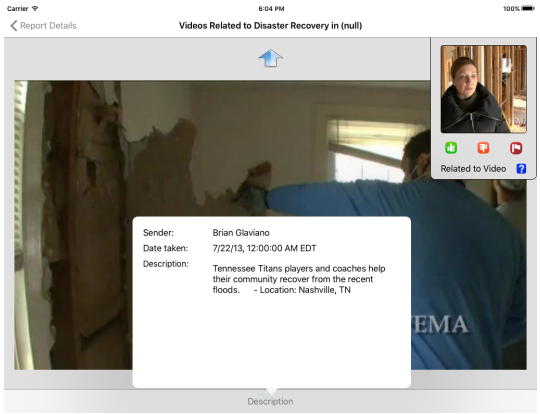
The system will be demonstrated via its front-end iPad application. The videos to be used in the demonstration were gathered from the Federal Emergency Management Agency (FEMA) website, which contains video contents not restricted to only the disaster event but also the disaster-related activities such as disaster beforehand preparation, disaster recovery, and training programs. The functionality presented to the users is described as follows.

The main page of the iPad application is a list of reports, which shows all the reports available in the system. Each row represents a specific report, and there are at most three small thumbnails shown on the right, which indicate the key frames of the videos associated with the report. Once a specific report is opened, the content of the report is shown

full-screen in the PDF format, and a side bar appears on the right of the screen which enables the users to see the list of associated videos as well as other information. The page also has a search function available for the users to locate the semantic connection between videos and the report (as shown in Figure 6.2a). The users can scroll through all the available videos listed in the tab view for the selected report, and press the play icon to play the video in the current position.



(a) List of videos for a report with highlighted keywords



(b) Video panel shows related videos for the subject "Disaster Recovery" with the video description button pressed

Figure 6.2: Screenshots from the iPad application.

Also, the user can tap any other place inside the video cell to bring up the timeline view of the video. The timeline is a set of videos that depict the same location and are organized by the dates from the earliest to the latest. For example, in the event of hurricane impact, the earliest videos in a certain timeline may show how the community prepares before the hurricane landfall, and the later videos may depict the disaster recovery process with some visible damages. In order to traverse the timeline of videos, the user can pan the screen by holding and dragging across the iPad's surface if the indicating arrows appear on the horizontal sides of the screen.

Furthermore, on the vertical sides, the indicating arrows represent the existing videos having high affinities with the one currently shown on the screen. That is, the videos that are classified into the same subject will appear when the user swipes the screen up or down. By holding the description button at the bottom of the video page, the text information about the current video will show, as illustrated in Figure 6.2b. The reference video will be shown as a thumbnail at the upper right corner, where the voting buttons are available for the users to interact with the system to provide feedback about the classification results.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

In this dissertation, a multimodal data analysis and fusion framework for data science is presented. It consists of three coherent components, namely, data analysis, multimodal deep representation learning, and multimodal fusion for semantic concept detection. These three components are integrated seamlessly and act as a coherent entity to provide essential functionalities in the proposed information discovery and analysis framework. More specifically:

- A Feature Affinity based Multiple Correspondence Analysis and Decision Fusion (FA-MCADF) framework is proposed to extract useful semantics from a disaster dataset. By utilizing the selected features and their affinities/ranks in each of the feature groups, the proposed framework is able to improve the concept detection results. Moreover, the decision fusion scheme further improves the accuracy performance.
- A framework of Multiple Correspondence Analysis based Neural Network (MCA-NN) is proposed to address the challenges in shallow learning. It integrates the Feature Affinity based Multiple Correspondence Analysis (FA-MCA) models into one large neural network model. The proposed semantic concept detection framework is utilized to classifiy the video-level concepts instead of frame-based images. Furthermore, the process of deciding the neural network module is automatic. The most important network parameters are obtained from the outputs of the FA-MCA models and the corresponding statistical information.

- A generalized framework is proposed to leverage the deep features from pre-trained CNN models in different applications by integrating EA and other techniques to automate the searching process. The proposed work determines the hyperparameters of a new neural network for one specific task after the best individual is selected. The model shows better performance than manually defined networks as it considers many characteristics of the datasets. Overall, the experimental results have proven that a time-consuming manual task could be done by an automated process that surpasses human ability and reaches an optimal solution effectively.
- A multimodal deep learning framework that incorporates sequential information from both audio and textual models is proposed to aid the disaster-related video classification. For the audio model, an effective and efficient deep learning model is utilized to extract the most discriminative and high-level features. The model is extended with a time-distributed fully connected layer and the subsequent LSTM layers. For the textual model, a pre-trained word embedding layer is used with a stacked LSTM model to generate the video-level concepts. Additionally, a novel two-stage fusion technique is proposed based on the frame-level image, audio, and video-level information by building a CNN model. Most notably, the image model predictions are incorporated into the audio model to adjust the classification ranking scores based on the reliability of the different predicted audio classes.
- A multimodal deep learning framework is proposed that utilizes different sources of information including visual, audio, and textual data. Unlike conventional fusion techniques (e.g., early fusion and late fusion), a two-stage modality fusion approach is proposed to first analyze the temporal information from both visual and audio data and then combine the textual information with the results from the first stage.

7.2 Future Work

In spite of the enormous efforts spent on the various tasks of multimodal data analytics and fusion, there is still much work to do in order to improve the current framework.

7.2.1 Semi-Supervised Learning for Multimedia Data Analytics

Over the past few years, researchers have developed innovative frameworks for multimedia information analytics [208, 209, 210]. Notably, the preprocessing steps for the classification of multimedia big data are tedious and time-consuming. Hence, in the future, we should focus on unsupervised or semi-supervised learning methods to automate the procedure or reduce the manual work This will make the process faster and more efeficient. With the ever-increasing popularity of social networks (e.g.,Twitter and Instagram), there has been a large growth in multimedia data such as images, posts, and video streaming generated by social media users. Over the time, the cumulative data must be better organized to efficiently utilize these resources. Additionally, the descriptive features that represent the multimedia data are normally large and difficult to manage. Nevertheless, some scientists point out that the features and data representations can be automatically separated by various ML algorithms without supervision [13, 150].

In the future, one of our objectives is to create a system that handles vast amounts of videos, images, and textual data by analyzing their correlations among the different concept levels with less supervision. Clustering techniques might be helpful to incorporate keyframe level classes into a video level concept. However, more experiments are needed to test if a system can automatically utilize the characteristics of the frame-level features into a video-level concept without manually labeling the relationships.

7.2.2 Automatic Deep Learning Model Selection and Construction for Multimodal Data Analytics

As mentioned in chapter 2, automated learning of neural network structures has been studied for many years. Specifically, researchers have paid significant attention to GA-based approaches to tune the network topology. Some initial efforts have focused on these directions; however, more work is needed to improve and evaluate the proposed solutions.

Extraordinary progress have been made by researchers in image classification mainly due to the accessibility of large-scale public visual datasets and powerful CNN models. Nevertheless, obtaining datasets as large as ImageNet for other classification tasks remains a challenging task [211]. To alleviate this problem, pre-trained models can be utilized for learning comprehensive features from smaller training datasets. It also supports the transfer of knowledge from one source domain to different target domains.

Currently, there are numerous frameworks to handle image classifications using transfer learning including preparing the preliminary features from the early layers of pre-trained CNN models, utilizing the mid-/high-level features, and fine-tuning the pre-trained CNN models to work for different targeting domains. With the purpose of building an intelligent framework to address various detection challenges, we need to carefully study the pre-trained models to understand the features that are extracted from each layer. During feature assessment, we can expect from 5,000 to one million combinations for one model. Each model differs in numerous ways depending on the number of layers. In such cases, examining the representation capabilities for certain layers in the pre-trained models is critical to identify the most relevant and useful features for different tasks.

Our future framework will assess a large number of sequences that demonstrate the feature selections of the pre-trained models across different modalities. The sequences created during the training stage are utilized in the evaluation stage for attaining the final

ranking scores. The ranking scores depict the reliability of different modalities. A higher score means a more discriminative model can be utilized for a specific task regarding a certain modality. The final training procedure commence when the contribute models are identified with the best performance given the specific task. The new model takes the normalized features as input to build the network that will generate the final prediction value for each test sets. Thus, the selection of the features highlights the similarities between the concepts from the pre-trained model and the target concepts. In some cases, features combined from multiple models might be more useful to create a constructive classifier rather than the features relies on limited modalities. In order to take advantage of the best representative features from the pre-trained models, the parameter optimization only occurs on the last training stage. Therefore, the testing stage simply uses a fixed model to compute the output. The same procedure applies for the conceptual assessment of the validation phase.

BIBLIOGRAPHY

[1] L. Zheng, C. Shen, L. Tang, C. Zeng, T. Li, S. Luis, and S.-C. Chen, "Data mining meets the needs of disaster information management," *IEEE Transactions on Human-Machine Systems*, vol. 43, pp. 451–464, 2013.

[2] D. Zhang, L. Zhou, and J. F. Nunamaker Jr, "A knowledge management framework for the support of decision making in humanitarian assistance/disaster relief," *Knowledge and Information Systems*, vol. 4, no. 3, pp. 370–385, 2002.

[3] Z. Ma, F. Nie, Y. Yang, J. R. Uijlings, N. Sebe, and A. G. Hauptmann, "Discriminating joint feature analysis for multimedia data understanding," *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1662–1672, 2012.

[4] S.-C. Chen, M.-L. Shyu, C. Zhang, and R. L. Kashyap, "Identifying overlapped objects for video indexing and modeling in multimedia database systems," *International Journal on Artificial Intelligence Tools*, vol. 10, no. 4, pp. 715–734, 2001.

[5] X. Huang, S.-C. Chen, M.-L. Shyu, and C. Zhang, "User concept pattern discovery using relevance feedback and multiple instance learning for content-based image retrieval," in *Third International Workshop on Multimedia Data Mining, in conjunction with the 8th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 100–108, July 2002.

[6] M. M. Mostafa, "More than words: social networks' text mining for consumer brand sentiments," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4241–4251, 2013.

[7] S. Vosoughi, P. Vijayaraghavan, and D. Roy, "Tweet2vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1041–1044, ACM, 2016.

[8] S.-C. Chen, M.-L. Shyu, M. Chen, and C. Zhang, "A decision tree-based multimodal data mining framework for soccer goal detection," in *IEEE International Conference on Multimedia and Expo*, pp. 265–268, 2004.

[9] A. Fleury, M. Vacher, and N. Noury, "SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results," *IEEE Trans. Information Technology in Biomedicine*, vol. 14, no. 2, pp. 274–283, 2010.

[10] D. Zhang, Y. Wang, L. Zhou, H. Yuan, and D. Shen, "Multimodal classification of Alzheimer's disease and mild cognitive impairment," *NeuroImage*, vol. 55, no. 3, pp. 856–867, 2011.

[11] L.-P. Morency, R. Mihalcea, and P. Doshi, "Towards multimodal sentiment analysis: harvesting opinions from the web," in *International Conference on Multimodal Interfaces*, pp. 169–176, ACM, 2011.

[12] M. Pantic, N. Sebe, J. F. Cohn, and T. Huang, "Affective multimodal human-computer interaction," in *ACM International Conference on Multimedia*, pp. 669–676, ACM, 2005.

[13] Q. Zhu, L. Lin, M.-L. Shyu, and S.-C. Chen, "Feature selection using correlation and reliability based scoring metric for video semantic detection," in *IEEE Fourth International Conference on Semantic Computing*, pp. 462–469, IEEE, 2010.

[14] Y. Yang, Z. Ma, A. G. Hauptmann, and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Transactions on Multimedia*, vol. 15, no. 3, pp. 661–669, 2013.

[15] K.-j. Kim and W. B. Lee, "Stock market prediction using artificial neural networks with optimal feature transformation," *Neural Computing & Applications*, vol. 13, no. 3, pp. 255–260, 2004.

[16] T. R. Reed and J. H. Dubuf, "A review of recent texture segmentation and feature extraction techniques," *CVGIP: Image Understanding*, vol. 57, no. 3, pp. 359–372, 1993.

[17] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive svms," in *15th ACM International Conference on Multimedia*, pp. 188–197, ACM, 2007.

[18] Q. Zhu, M.-L. Shyu, and S.-C. Chen, "Discriminative learning-assisted video semantic concept classification," *Multimedia Security: Watermarking, Steganography, and Forensics*, p. 31, 2012.

[19] L. Xie and A. L. Yuille, "Genetic CNN," *CoRR*, vol. abs/1703.01513, 2017.

[20] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *British Machine Vision Conference*, pp. 124.1–124.11, BMVA Press, 2009.

[21] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue, "Modeling spatial-temporal clues in a hybrid deep learning framework for video classification," in *The 23rd ACM international conference on Multimedia*, pp. 461–470, ACM, 2015.

[22] J.-T. Tsai, J.-H. Chou, and T.-K. Liu, "Tuning the structure and parameters of a neural network by using hybrid taguchi-genetic algorithm," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 69–80, 2006.

[23] S. R. Young, D. C. Rose, T. P. Karnowski, S.-H. Lim, and R. M. Patton, "Optimizing deep learning hyper-parameters through an evolutionary algorithm," in *The Workshop on Machine Learning in High-Performance Computing Environments*, pp. 4:1–4:5, ACM, 2015.

[24] E. P. Ijjina and K. M. Chalavadi, "Human action recognition using genetic algorithms and convolutional neural networks," *Pattern Recognition*, vol. 59, pp. 199–212, 2016.

[25] P. O. Nunally and D. R. MacCormack, "Multimedia data analysis in intelligent video information management system," Mar. 7 2000. US Patent 6,035,341.

[26] S.-C. Chen, S. Sista, M.-L. Shyu, and R. L. Kashyap, "Augmented transition networks as video browsing models for multimedia databases and multimedia information systems," in *The 11th IEEE International Conference on Tools with Artificial Intelligence*, pp. 175–182, IEEE, 1999.

[27] H. A. Elnemr, N. M. Zayed, and M. A. Fakhreldein, "Feature extraction techniques: Fundamental concepts and survey," *Handbook of Research on Emerging Perspective in Intelligent Pattern Recognition Analysis and Image Processing*, 2015.

[28] M.-L. Shyu, K. Sarinnapakorn, I. Kuruppu-Appuhamilage, S.-C. Chen, L. Chang, and T. Goldring, "Handling nominal features in anomaly intrusion detection problems," in *The 15th International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications*, pp. 55–62, 2005.

[29] J. Fan, H. Luo, J. Xiao, and L. Wu, "Semantic video classification and feature subset selection under context and concept uncertainty," in *The 4th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 192–201, ACM, 2004.

[30] L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen, "Effective feature space reduction with imbalanced data for semantic concept detection," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp. 262–269, 2008.

[31] S.-C. Chen, M.-L. Shyu, C. Zhang, and M. Chen, "A multimodal data mining framework for soccer goal detection based on decision tree logic," *International Journal of Computer Applications in Technology*, vol. 27, pp. 312–323, 2006.

[32] D. Liu, Y. Yan, M.-L. Shyu, G. Zhao, and M. Chen, "Spatio-temporal analysis for human action detection and recognition in uncontrolled environments," *International Journal of Multimedia Data Engineering and Management*, vol. 6, no. 1, pp. 1–18, 2015.

[33] L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen, "Video semantic concept discovery using multimodal-based association classification," in *The IEEE International Conference on Multimedia and Expo*, pp. 859–862, 2007.

[34] B. Heisele, T. Serre, S. Prentice, and T. Poggio, "Hierarchical classification and feature reduction for fast face detection with support vector machines," *Pattern Recognition*, vol. 36, no. 9, pp. 2007–2017, 2003.

[35] Y. Wu, E. Y. Chang, K. C.-C. Chang, and J. R. Smith, "Optimal multimodal fusion for multimedia data analysis," in *The 12th annual ACM International Conference on Multimedia*, pp. 572–579, ACM, 2004.

[36] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 2, no. 1, pp. 1–19, 2006.

[37] Y. Lu, I. Cohen, X. S. Zhou, and Q. Tian, "Feature selection using principal feature analysis," in *The 15th ACM International Conference on Multimedia*, pp. 301–304, ACM, 2007.

[38] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems Man and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[39] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[40] L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen, "Correlation-based video semantic concept detection using multiple correspondence analysis," in *Tenth IEEE International Symposium on Multimedia*, pp. 316–321, IEEE, 2008.

[41] L. Lin and M.-L. Shyu, "Weighted association rule mining for video semantic detection," *Methods and Innovations for Multimedia Database Content Management*, vol. 1, no. 1, pp. 37–54, 2012.

[42] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, *et al.*, "Wide & deep learning for recommender systems," in *The 1st Workshop on Deep Learning for Recommender Systems*, pp. 7–10, ACM, 2016.

[43] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, "Deep learning for answer sentence selection," *CoRR*, vol. abs/1412.1632, 2014.

[44] M. Tan, B. Xiang, and B. Zhou, "LSTM-based deep learning models for non-factoid answer selection," *CoRR*, vol. abs/1511.04108, 2015.

[45] S. Liao, Y. Gao, A. Oto, and D. Shen, "Representation learning: a unified deep learning framework for automatic prostate mr segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 254–261, Springer, 2013.

[46] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5987–5995, IEEE, 2017.

[47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *The IEEE Cconference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[48] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *AAAI Conference on Artificial Intelligence*, pp. 4278–4284, AAAI Press, 2017.

[49] S. Pouyanfar and S.-C. Chen, "Automatic video event detection for imbalance data using enhanced ensemble deep learning," *International Journal of Semantic Computing*, vol. 11, no. 01, pp. 85–109, 2017.

[50] M. Papakostas, E. Spyrou, T. Giannakopoulos, G. Siantikos, D. Sgouropoulos, P. Mylonas, and F. Makedon, "Deep visual attributes vs. hand-crafted audio features on multidomain speech emotion recognition," *Computation*, vol. 5, no. 2, p. 26, 2017.

[51] D. H. Wolpert, "The supervised learning no-free-lunch theorems," in *Soft Computing and Industry*, pp. 25–42, Springer, 2002.

[52] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[54] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[55] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *The IEEE conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[57] T. Back, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[58] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," in *Feature Extraction, Construction and Selection*, pp. 117–136, Springer, 1998.

[59] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary computation*, vol. 3, no. 2, pp. 82–102, 1999.

[60] P. Lameski, E. Zdravevski, R. Mingov, and A. Kulakov, "Svm parameter tuning with grid search and its impact on reduction of model over-fitting," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, pp. 464–474, Springer, 2015.

[61] H. Mania, A. Guy, and B. Recht, "Simple random search provides a competitive approach to reinforcement learning," *CoRR*, vol. abs/1803.07055, 2018.

[62] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[63] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, pp. 2951–2959, 2012.

[64] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *The 34th International Conference on Machine Learning*, vol. 70, pp. 2208–2217, JMLR. org, 2017.

[65] H. Tian, S. Pouyanfar, J. Chen, S.-C. Chen, and S. S. Iyengar, "Automatic convolutional neural network selection for image classification using genetic algorithms," in *The IEEE International Conference on Information Reuse and Integration*, pp. 444–451, IEEE, 2018.

[66] S. R. Young, D. C. Rose, T. P. Karnowski, S.-H. Lim, and R. M. Patton, "Optimizing deep learning hyper-parameters through an evolutionary algorithm," in *The Workshop on Machine Learning in High-Performance Computing Environments*, pp. 4:1–4:5, ACM, 2015.

[67] F. H.-F. Leung, H.-K. Lam, S.-H. Ling, and P. K.-S. Tam, "Tuning of the structure and parameters of a neural network using an improved genetic algorithm," *IEEE Transactions on Neural networks*, vol. 14, no. 1, pp. 79–88, 2003.

[68] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems*, pp. 2546–2554, 2011.

[69] O. E. David and I. Greental, "Genetic algorithms for evolving deep neural networks," in *The Annual Conference on Genetic and Evolutionary Computation*, pp. 1451–1452, ACM, 2014.

[70] M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," in *The Genetic and Evolutionary Computation Conference*, pp. 497–504, ACM, 2017.

[71] H. Lam, S. Ling, F. H. Leung, and P. K.-S. Tam, "Tuning of the structure and parameters of neural network using an improved genetic algorithm," in *The 27th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, pp. 25–30, IEEE, 2001.

[72] D. Hossain, G. Capi, and M. Jindai, "Optimizing deep learning parameters using genetic algorithm for object recognition and robot grasping," *Journal of Electronic Science and Technology*, vol. 16, no. 1, pp. 11–15, 2018.

[73] L. Xie and A. Yuille, "Genetic cnn," in *The IEEE International Conference on Computer Vision*, pp. 1379–1388, 2017.

[74] I. Loshchilov and F. Hutter, "CMA-ES for hyperparameter optimization of deep neural networks," *CoRR*, vol. abs/1604.07269, 2016.

[75] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43–62, 1997.

[76] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[77] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Deep learning identity-preserving face space," in *The IEEE International Conference on Computer Vision*, pp. 113–120, IEEE, 2013.

[78] Y. Kim, "Convolutional neural networks for sentence classification," in *Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, ACL, 2014.

[79] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 273–278, IEEE, 2013.

[80] T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "Pcanet: A simple deep learning baseline for image classification?," *IEEE Trans. Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[81] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[82] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10, p. 1995, 1995.

[83] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[84] H. Cui, H. Zhang, G. R. Ganger, P. B. Gibbons, and E. P. Xing, "Geeps: Scalable deep learning on distributed gpus with a gpu-specialized parameter server," in *The Eleventh European Conference on Computer Systems*, p. 4, ACM, 2016.

[85] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov, "Applications of deep learning in biomedicine," *Molecular Pharmaceutics*, vol. 13, no. 5, pp. 1445–1454, 2016.

[86] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International Conference on Machine Learning*, pp. 647–655, 2014.

[87] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[88] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

[89] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, ACL, 2014.

[90] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *CoRR*, vol. abs/1312.6026, 2013.

[91] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *The Thirteenth International Conference on Artificial Intelligence and Statistics*, vol. 9, pp. 249–256, JMLR.org, 2010.

[92] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4520–4524, IEEE, 2015.

[93] S.-C. Chen, M.-L. Shyu, and R. Kashyap, "Augmented transition network as a semantic model for video data," *International Journal of Networking and Information Systems*, vol. 3, no. 1, pp. 9–25, 2000.

[94] S.-C. Chen, M.-L. Shyu, and C. Zhang, "Innovative shot boundary detection for video indexing," in *Video Data Management and Information Retrieval* (S. Deb, ed.), pp. 217–236, Idea Group Publishing, 2005.

[95] X. Chen, C. Zhang, S.-C. Chen, and S. Rubin, "A human-centered multiple instance learning framework for semantic video retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 39, no. 2, pp. 228–233, 2009.

[96] T. Meng and M.-L. Shyu, "Leveraging concept association network for multimedia rare concept mining and retrieval," in *The IEEE International Conference on Multimedia and Expo*, July 2012.

[97] S.-C. Chen, M.-L. Shyu, C. Zhang, and M. Chen, "A multimodal data mining framework for soccer goal detection based on decision tree logic," *International Journal of Computer Applications in Technology*, vol. 27, no. 4, p. 312, 2006.

[98] S.-C. Chen and R. L. Kashyap, "A spatio-temporal semantic model for multimedia database systems and multimedia information systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 4, pp. 607–622, 2001.

[99] B. Chen, *Deep learning of invariant spatio-temporal features from video*. PhD thesis, University of British Columbia, 2010.

[100] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

[101] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.

[102] T. Meng, L. Lin, M.-L. Shyu, and S.-C. Chen, "Histology image classification using supervised classification and multimodal fusion," in *IEEE International Symposium on Multimedia*, pp. 145–152, IEEE, 2010.

[103] C. Haruechaiyasak, M.-L. Shyu, and S.-C. Chen, "Web document classification based on fuzzy association," in *The 26th Annual International Computer Software and Applications*, pp. 487–492, IEEE, 2002.

[104] R. Potharaju, B. Carbunar, M. Azimpourkivi, V. Vasudevan, and S. Iyengar, "Infiltrating social network accounts: attacks and defenses," in *Secure System Design and Trustable Computing*, pp. 457–485, Springer, 2016.

[105] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter," in *International Workshop on Semantic Evaluation*, pp. 502–518, 2017.

[106] C. Chen, Q. Zhu, L. Lin, and M.-L. Shyu, "Web media semantic concept retrieval via tag removal and model fusion," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, p. 61, 2013.

[107] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia Syst.*, vol. 16, no. 6, pp. 345–379, 2010.

[108] M. Chen, S.-C. Chen, M.-L. Shyu, and C. Zhang, "Video event mining via multimodal content analysis and classification," in *Multimedia Data Mining and Knowledge Discovery*, pp. 234–258, Springer, 2007.

[109] H. Shahbazi, K. Jamshidi, A. H. Monadjemi, and H. E. Manoochehri, "Training oscillatory neural networks using natural gradient particle swarm optimization," *Robotica*, vol. 33, no. 7, pp. 1551–1567, 2015.

[110] H. Xue, Y. Liu, D. Cai, and X. He, "Tracking people in rgbd videos using deep learning and motion clues," *Neurocomputing*, vol. 204, pp. 70–76, 2016.

[111] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014.

[112] R. Johnson and T. Zhang, "Supervised and semi-supervised text categorization using lstm for region embeddings," in *International Conference on Machine Learning*, pp. 526–534, JMLR.org, 2016.

[113] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski, *et al.*, "Emonets: multimodal deep learning approaches for emotion recognition in video," *Journal on Multimodal User Interfaces*, vol. 10, no. 2, pp. 99–111, 2016.

[114] S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. S. Iyengar, "Multimedia big data analytics: A survey," *ACM Computing Surveys*, vol. 51, no. 1, pp. 10:1–10:34, 2018.

[115] W. Zhu, P. Cui, Z. Wang, and G. Hua, "Multimedia big data computing," *IEEE Multimedia*, vol. 22, no. 3, pp. 96–c3, 2015.

[116] C. Chen, Q. Zhu, L. Lin, and M.-L. Shyu, "Web media semantic concept retrieval via tag removal and model fusion," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, pp. 61:1–61:22, October 2013.

[117] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *IEEE International Conference on Computer Vision*, pp. 2722–2730, 2015.

[118] H. Shahbazi, K. Jamshidi, A. H. Monadjemi, and H. Eslami, "Biologically inspired layered learning in humanoid robots," *Knowledge-Based Systems*, vol. 57, pp. 8–27, 2014.

[119] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng, "Early diagnosis of alzheimer's disease with deep learning," in *IEEE 11th International Symposium on Biomedical Imaging*, pp. 1015–1018, IEEE, 2014.

[120] T. Meng, A. T. Soliman, M.-L. Shyu, Y. Yang, S.-C. Chen, S. Iyengar, J. S. Yordy, and P. Iyengar, "Wavelet analysis in current cancer genome research: a survey," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 6, pp. 1442–14359, 2013.

[121] H. Tian, H. C. Zheng, and S.-C. Chen, "Sequential deep learning for disaster-related video classification," in *The First IEEE International Conference on Multimedia Information Processing and Retrieval*, pp. 106–111, 2018.

[122] M. E. P. Reyes, S. Pouyanfar, H. C. Zheng, H.-Y. Ha, and S.-C. Chen, "Multimedia data management for disaster situation awareness," in *International Symposium on Sensor Networks, Systems and Security*, pp. 137–146, Springer, 2017.

[123] L. Zheng, C. Shen, L. Tang, C. Zeng, T. Li, S. Luis, and S.-C. Chen, "Data mining meets the needs of disaster information management," *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 5, pp. 451–464, 2013.

[124] S.-C. Chen, M. Chen, N. Zhao, S. Hamid, K. Chatterjee, and M. Armella, "Florida public hurricane loss model: Research in multi-disciplinary system integration assisting government policy making," *Government Information Quarterly*, vol. 26, no. 2, pp. 285–294, 2009.

[125] H. Tian and S.-C. Chen, "A video-aided semantic analytics system for disaster information integration," in *IEEE International Conference on Multimedia Big Data*, pp. 242–243, 2017.

[126] L. Zheng, C. Shen, L. Tang, T. Li, S. Luis, S.-C. Chen, and V. Hristidis, "Using data mining techniques to address critical information exchange needs in disaster affected public-private networks," in *The 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 125–134, ACM, 2010.

[127] N. Rishe, J. Yuan, R. Athauda, S.-C. Chen, X. Lu, X. Ma, A. Vaschillo, A. Shaposhnikov, and D. Vasilevsky, "Semantic access: semantic interface for querying databases," in *VLDB*, pp. 591–594, 2000.

[128] L. Zheng, C. Shen, L. Tang, T. Li, S. Luis, and S.-C. Chen, "Applying data mining techniques to address disaster information management challenges on mobile devices," in *The 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 283–291, ACM, 2011.

[129] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Surveys*, vol. 51, no. 5, p. 92, 2018.

[130] D. Wang and T. F. Zheng, "Transfer learning for speech and language processing," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1225–1237, IEEE, 2015.

[131] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient transfer learning," *CoRR*, vol. abs/1611.06440, 2016.

[132] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[133] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[134] L. Deng, D. Yu, *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[135] H.-Y. Ha, Y. Yang, S. Pouyanfar, H. Tian, and S.-C. Chen, "Correlation-based deep learning for multimedia semantic concept detection," in *International Conference on Web Information Systems Engineering*, pp. 473–487, 2015.

[136] Y. Yang, W. Lu, J. Domack, T. Li, S.-C. Chen, S. Luis, and J. K. Navlakha, "MADIS: A multimedia-aided disaster information integration system for emergency management," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 233–241, IEEE, 2012.

[137] L. Bruzzone and D. F. Prieto, "A technique for the selection of kernel-function parameters in rbf neural networks for classification of remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 1179–1184, 1999.

[138] X. Liao, Y. Xue, and L. Carin, "Logistic regression with an auxiliary data source," in *The 22nd International Conference on Machine Learning*, pp. 505–512, ACM, 2005.

[139] J. S. Boreczky and L. A. Rowe, "Comparison of video shot boundary detection techniques," *Journal of Electronic Imaging*, vol. 5, no. 2, pp. 122–128, 1996.

[140] S.-C. Chen, M.-L. Shyu, and C. Zhang, "Innovative shot boundary detection for video indexing," *Video Data Management and Information Retrieval*, pp. 217–236, 2005.

[141] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, IEEE, 2005.

[142] S. A. Chatzichristofis and Y. S. Boutalis, "Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval," in *International Conference on Computer Vision Systems*, pp. 312–322, Springer, 2008.

[143] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *International Conference on Image Processing*, vol. 1, pp. I–900, IEEE, 2002.

[144] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the hsv color space for image retrieval," in *The 2002 International Conference on Image Processing*, vol. 2, pp. II–589, IEEE, 2002.

[145] X. Li, S.-C. Chen, M.-L. Shyu, and B. Furht, "Image retrieval by color, texture, and spatial information," in *Proceedings of the 8th International Conference on Distributed Multimedia Systems*, pp. 152–159, September 2002.

[146] N. Takahashi, M. Gygli, and L. V. Gool, "AENet: Learning deep audio features for video analysis," *CoRR*, vol. abs/1701.00599, 2017.

[147] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems*, pp. 892–900, 2016.

[148] M.-L. Shyu, S.-C. Chen, and C. Haruechaiyasak, "Mining user access behavior on the www," in *The IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 1717–1722, IEEE, 2001.

[149] S.-C. Chen, "Multimedia databases and data management: a survey," *International Journal of Multimedia Data Engineering and Management*, vol. 1, no. 1, pp. 1–11, January-March 2010.

[150] M.-L. Shyu, S.-C. Chen, and R. L. Kashyap, "Generalized affinity-based association rule mining for multimedia database queries," *Knowledge and Information Systems*, vol. 3, no. 3, pp. 319–337, 2001.

[151] Q. Zhu, L. Lin, M.-L. Shyu, and S.-C. Chen, "Effective supervised discretization for classification based on correlation maximization," in *IEEE International Conference on Information Reuse and Integration*, pp. 390–395, 2011.

[152] L. Lin, M.-L. Shyu, G. Ravitz, and S.-C. Chen, "Video semantic concept detection via associative classification," in *The IEEE International Conference on Multimedia and Expo*, pp. 418–421, IEEE, 2009.

[153] X. Chen, C. Zhang, S.-C. Chen, and M. Chen, "A latent semantic indexing based method for solving multiple instance learning problem in region-based image retrieval," in *The Seventh IEEE International Symposium on Multimedia*, pp. 37–44, Dec 2005.

[154] S.-C. Chen and R. Kashyap, "Temporal and spatial semantic models for multimedia presentations," in *International Symposium on Multimedia Information Processing*, pp. 441–446, 1997.

[155] S.-C. Chen, M.-L. Shyu, and C. Zhang, "An intelligent framework for spatio-temporal vehicle tracking," in *The 4th IEEE International Conference on Intelligent Transportation Systems*, pp. 213–218, August 2001.

[156] H. Vafaie and I. F. Imam, "Feature selection methods: genetic algorithms vs. greedy-like search," in *The International Conference on Fuzzy and Intelligent Control Systems*, vol. 51, 1994.

[157] Y. Yang, H.-Y. Ha, F. Fleites, S.-C. Chen, and S. Luis, "Hierarchical disaster image classification for situation report enhancement," in *The IEEE International Conference on Information Reuse and Integration*, pp. 181–186, IEEE, 2011.

[158] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *The International Joint Conference on Artificial Intelligence*, vol. 14, pp. 1137–1145, 1995.

[159] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *International Journal of Machine Learning Technology*, 2011.

[160] G. Holmes, A. Donkin, and I. H. Witten, "WEKA: A machine learning workbench," in *The Second Australian and New Zealand Conference on Intelligent Information Systems*, pp. 357–361, IEEE, 1994.

[161] S.-C. Chen, R. L. Kashyap, and A. Ghafoor, *Semantic models for multimedia database searching and browsing*, vol. 21. Springer Science & Business Media, 2000.

[162] M.-L. Shyu, C. Haruechaiyasak, and S.-C. Chen, "Category cluster discovery from distributed www directories," *Information Sciences*, vol. 155, no. 3, pp. 181–197, 2003.

[163] Y. Yang, H.-Y. Ha, F. C. Fleites, and S.-C. Chen, "A multimedia semantic retrieval mobile system based on hcfgs," *IEEE MultiMedia*, vol. 21, no. 1, pp. 36–46, 2014.

[164] H.-Y. Ha, F. C. Fleites, S.-C. Chen, and M. Chen, "Correlation-based re-ranking for semantic concept detection," in *IEEE 15th International Conference on Information Reuse and Integration*, pp. 765–770, IEEE, 2014.

[165] L. Lin, M.-L. Shyu, and S.-C. Chen, "Rule-based semantic concept classification from large-scale video collections," *International Journal of Multimedia Data Engineering and Management*, vol. 4, no. 1, pp. 46–67, 2013.

[166] M.-L. Shyu, S.-C. Chen, M. Chen, C. Zhang, and K. Sarinnapakorn, "Image database retrieval utilizing affinity relationships," in *The 1st ACM International Workshop on Multimedia Databases*, pp. 78–85, ACM, 2003.

[167] L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen, "Video semantic concept discovery using multimodal-based association classification," in *IEEE International Conference on Multimedia and Expo*, pp. 859–862, IEEE, 2007.

[168] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, pp. 3320–3328, 2014.

[169] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *The 32nd International Conference on Machine Learning*, pp. 97–105, 2015.

[170] G. Awad, C. G. Snoek, A. F. Smeaton, and G. Quénot, "TRECVid semantic indexing of video: A 6-year retrospective," *ITE Transactions on Media Technology and Applications*, vol. 4, no. 3, pp. 187–208, 2016.

[171] C. M. Anderson-Cook, *Practical genetic algorithms*. Taylor & Francis, 2005.

[172] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[173] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[174] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[175] H. Tian, Y. Tao, S. Pouyanfar, S.-C. Chen, and M.-L. Shyu, "Multimodal deep representation learning for video classification," *World Wide Web*, vol. 22, no. 3, pp. 1325–1341, 2019.

[176] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y.-H. Lu, S.-C. Chen, and M.-L. Shyu, "Dynamic sampling

in convolutional neural networks for imbalanced data classification," in *The First IEEE International Conference on Multimedia Information Processing and Retrieval*, pp. 112–117, 2018.

[177] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4694–4702, 2015.

[178] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[179] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[180] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," *CoRR*, vol. abs/1611.02167, 2016.

[181] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, *et al.*, "Evolving deep neural networks," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pp. 293–312, Elsevier, 2019.

[182] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *CoRR*, vol. abs/1611.01578, 2016.

[183] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[184] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette, "Evolutionary algorithms for reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 11, pp. 241–276, 1999.

[185] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, "Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning," *CoRR*, vol. abs/1712.06567, 2017.

[186] S. Whiteson and P. Stone, "Evolutionary function approximation for reinforcement learning," *Journal of Machine Learning Research*, vol. 7, no. May, pp. 877–917, 2006.

[187] Y. Tian, S.-C. Chen, M.-L. Shyu, T. Huang, P. Sheu, and A. Del Bimbo, "Multimedia big data," *IEEE MultiMedia*, vol. 22, no. 3, pp. 93–95, 2015.

[188] Y. Yan, Q. Zhu, M.-L. Shyu, and S.-C. Chen, "Classifier fusion by judgers on spark clusters for multimedia big data classification," in *Quality Software Through Reuse and Integration*, pp. 91–108, Cham: Springer, 2016.

[189] Z. Lan, L. Bao, S. Yu, W. Liu, and A. G. Hauptmann, "Multimedia classification and event detection using double fusion," *Multimedia Tools and Applications*, vol. 71, no. 1, pp. 333–347, 2014.

[190] T. Meng and M.-L. Shyu, "Leveraging concept association network for multimedia rare concept mining and retrieval," in *IEEE International Conference on Multimedia and Expo*, pp. 860–865, 2012.

[191] M.-L. Shyu, S.-C. Chen, and R. L. Kashyap, "Augmented transition network as a semantic model for video data," *International Journal of Networking and Information Systems, Special Issue on Video Data*, vol. 3, no. 1, pp. 9–25, 2000.

[192] J. Scott, *Social network analysis*. SAGE, 2017.

[193] T. Li, N. Xie, C. Zeng, W. Zhou, L. Zheng, Y. Jiang, Y. Yang, H. Ha, W. Xue, Y. Huang, S. Chen, J. K. Navlakha, and S. S. Iyengar, "Data-driven techniques in disaster information management," *ACM Computing Surveys*, vol. 50, no. 1, p. 1, 2017.

[194] S. Pouyanfar and S.-C. Chen, "Semantic concept detection using weighted discretization multiple correspondence analysis for disaster information management," in *International Conference on Information Reuse and Integration*, pp. 556–564, 2016.

[195] H. Tian, S.-C. Chen, S. H. Rubin, and W. K. Grefe, "FA-MCADF: Feature affinity based multiple correspondence analysis and decision fusion framework for disaster information management," in *IEEE International Conference on Information Reuse and Integration*, pp. 198–206, 2017.

[196] Y. Yang, S. Pouyanfar, H. Tian, M. Chen, S.-C. Chen, and M.-L. Shyu, "IF-MCA: Importance factor-based multiple correspondence analysis for multimedia data analytics," *IEEE Transactions on Multimedia*, 2017.

[197] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[198] H. Tian and S.-C. Chen, "MCA-NN: Multiple correspondence analysis based neural network for disaster information detection," in *IEEE International Conference on Multimedia Big Data*, pp. 268–275, 2017.

[199] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[200] W. J. Reichmann, *Use and abuse of statistics*. Penguin books, 1964.

[201] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[202] M.-L. Shyu, C. Haruechaiyasak, S.-C. Chen, and N. Zhao, "Collaborative filtering by mining association rules from user access sequences," in *The International Workshop on Challenges in Web Information Retrieval and Integration*, pp. 128–135, April 2005.

[203] M.-L. Shyu, T. Quirino, Z. Xie, S.-C. Chen, and L. Chang, "Network intrusion detection through adaptive sub-eigenspace modeling in multiagent systems," *ACM Trans. Auton. Adapt. Syst.*, vol. 2, Sept. 2007.

[204] M.-L. Shyu, C. Haruechaiyasak, and S.-C. Chen, "Category cluster discovery from distributed www directories," *Journal of Information Sciences*, vol. 155, pp. 181–197, 2003.

[205] S. Luis, F. C. Fleites, Y. Yang, H.-Y. Ha, and S.-C. Chen, "A visual analytics multimedia mobile system for emergency response," in *IEEE International Symposium on Multimedia*, pp. 337–338, IEEE, 2011.

[206] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "A framework and graphical development environment for robust nlp tools and applications.," in

the 40th Anniversary Meeting of the Association for Computational Linguistics, pp. 168–175, 2002.

[207] H. Tian and S.-C. Chen, "MCA-NN: Multiple correspondence analysis based neural network for disaster information detection," in *IEEE International Conference on Multimedia Big Data*, pp. 268–275, 2017.

[208] M.-L. Shyu, S.-C. Chen, M. Chen, and C. Zhang, "A unified framework for image database clustering and content-based retrieval," in *The 2nd ACM International Workshop on Multimedia Databases*, pp. 19–27, ACM, 2004.

[209] X. Chen, C. Zhang, S.-C. Chen, and M. Chen, "A latent semantic indexing based method for solving multiple instance learning problem in region-based image retrieval," in *The Seventh IEEE International Symposium on Multimedia*, pp. 8–pp, IEEE, 2005.

[210] S.-C. Chen, S. H. Rubin, M.-L. Shyu, and C. Zhang, "A dynamic user concept pattern learning framework for content-based image retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 36, no. 6, pp. 772–783, 2006.

[211] Y. Yan, M. Chen, M.-L. Shyu, and S.-C. Chen, "Deep learning for imbalanced multimedia data classification," in *IEEE International Symposium on Multimedia*, pp. 483–488, 2015.

VITA

HAIMAN TIAN

October 1, 1986	Born, Beijing, China
2005–2009	B.S., Computer Science Sun Yat-Sen University Guangzhou, Guangdong, China
2013–2014	M.S., Computer Engineering Florida International University Miami, Florida
2014–2019	Ph.D., Computer Science Florida International University Miami, Florida

PUBLICATIONS

Haiman Tian, Yudong Tao, Samira Pouyanfar, Shu-Ching Chen, Mei-Ling Shyu, "Multimodal Deep Representation Learning for Video Classification," *World Wide Web*, 22(3), pp. 1325-1341, 2018.

Yimin Yang, Samira Pouyanfar, Haiman Tian, Min Chen, Shu-Ching Chen, and Mei-Ling Shyu, "IF-MCA: Importance Factor-based Multiple Correspondence Analysis for Multimedia Data Analytics," *IEEE Transactions on Multimedia*, Volume 20, Issue 4, pp. 1024-1032, April 2018.

Samira Pouyanfar, Yudong Tao, Haiman Tian, Shu-Ching Chen, Mei-Ling Shyu, "Multimodal Deep Learning based on Multiple Correspondence Analysis for Disaster Management," *World Wide Web*, pp. 1-19, 2018.

Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, S. S. Iyengar, "A Survey on Deep Learning: Algorithms, Techniques, and Applications," *ACM Computing Surveys (CSUR)*, 51, no. 5 (2018): 92.

Haiman Tian, Samira Pouyanfar, Jonathan Chen, Shu-Ching Chen and Sitharama S. Iyengar, "Automatic Convolutional Neural Network Selection for Image Classification using Genetic Algorithm," *The 19th IEEE International Conference on Information Reuse and Integration (IRI 2018)*, pp. 444-451, 2018.

Samira Pouyanfar, Yudong Tao, Anup Mohan, Haiman Tian, Ahmed S. Kaseb, Kent Gauen, Ryan Dailey, Sarah Aghajanzadeh, Yung-Hsiang Lu, Shu-Ching Chen, Mei-Ling

Shyu, "Dynamic Sampling in Convolutional Neural Networks for Imbalanced Data Classification," *The First IEEE International Conference on Multimedia Information Processing and Retrieval (IEEE MIPR 2018)*, pp. 112-117, 2018.

Haiman Tian, Hector Cen Zheng, Shu-Ching Chen, "Sequential Deep Learning for Disaster-Related Video Classification," *The First IEEE International Conference on Multimedia Information Processing and Retrieval (IEEE MIPR 2018)*, pp. 106-111, 2018.

Haiman Tian, Shu-Ching Chen, Stuart H. Rubin, and William K. Grefe, "FA-MCADF: Feature Affinity based Multiple Correspondence Analysis and Decision Fusion Framework for Disaster Information Management," *IEEE 18th International Conference on Information Reuse and Integration (IEEE IRI 2017)*, pp. 198-206, 2017.

Haiman Tian and Shu-Ching Chen, "MCA-NN: Multiple Correspondence Analysis based Neural Network for Disaster Information Detection," *The Third IEEE International Conference on Multimedia Big Data (IEEE BigMM 2017)*, pp. 268-275, 2017.

Haiman Tian and Shu-Ching Chen, "A Video-aided Semantic Analytics System for Disaster Information Integration," *The Third IEEE International Conference on Multimedia Big Data (IEEE BigMM 2017)*, pp. 242-243, 2017.

Yilin Yan, Samira Pouyanfar, Haiman Tian, Sheng Guan, Hsin-Yu Ha, Shu-Ching Chen, Mei-Ling Shyu, and Shahid Hamid, "Domain Knowledge Assisted Data Processing for Florida Public Hurricane Loss Model," *The 17th IEEE International Conference on Information Reuse and Integration (IRI 2016)*, pp. 441-447, 2016.

Haiman Tian, Hsin-Yu Ha, Samira Pouyanfar, Yilin Yan, Sheng Guan, Shu-Ching Chen, Mei-Ling Shyu, and Shahid Hamid, "A Scalable and Automatic Validation Process for Florida Public Hurricane Loss Model," *The 17th IEEE International Conference on Information Reuse and Integration (IRI 2016)*, pp. 324-331, 2016.

Hsin-Yu Ha, Yimin Yang, Samira Pouyanfar, Haiman Tian, and Shu-Ching Chen, "Correlation-based Deep Learning for Multimedia Semantic Concept Detection," *The 16th International Conference on Web Information System Engineering (WISE 2015)*, pp. 473-487, 2015

Yimin Yang, Daniel Lopez, Haiman Tian, Samira Pouyanfar, Fausto Fleites, Shu-Ching Chen and Shahid Hamid, "Integrated Execution Framework for Catastrophe Modeling," *Ninth IEEE International Conference on Semantic Computing (ICSC2015)*, pp. 201-207, 2015.

Samira Pouyanfar, Yudong Tao, Haiman Tian, Maria Presa Reyes, Yuexuan Tu, Yilin Yan, Tianyi Wang, Hector Cen, Yingxin Li, Saad Sadiq, Mei-Ling Shyu, Shu-Ching Chen, Winnie Chen, Tiffany Chen, Jonathan Chen, "Florida International University-University of Miami TRECVID 2018," 2018.