SCTR's Pune Institute of Computer Technology
Dhankawadi, Pune

AN INTERNSHIP REPORT ON

Research Internship in Image Classification

SUBMITTED BY
Name: Sagar Abhyankar
Class: TE 01
Roll no: 31102

Under the guidance of
Prof. Sarang Joshi

DEPARTMENT OF COMPUTER ENGINEERING
ACADEMIC YEAR 2022-23

DEPARTMENT OF COMPUTER ENGINEERING

SCTR's Pune Institute of Computer Technology
Dhankawadi, Pune
Maharashtra 411043

CERTIFICATE

This is to certify that the SPPU Curriculum-based internship report entitled
"Research Internship in Image Classification"

Submitted by
Sagar Abhyankar
(Exam No. 31102)

has satisfactorily completed the curriculum-based internship under the guidance of *Prof. Sarang Joshi* towards the partial fulfillment of third year Computer Engineering Semester VI, Academic Year 2022-23 of Savitribai Phule Pune University.

Prof. Sarang Joshi
Internship Guide
PICT, Pune

Dr. G. V. Kale
Head
Department of Computer Engineering
PICT, Pune

Place:Pune
Date: 15/05/2023

Acknowledgement

It gives me great pleasure in presenting the internship report on "Research internship in Image classification".

First of all I would like to take this opportunity to thank my internship guide Prof. Sarang Joshi for giving me all the help and guidance needed. I am really grateful for his kind support and valuable suggestions that proved to be beneficial in the overall completion of this internship.

I am thankful to our Head of Computer Engineering Department, Dr. G.V.Kale, for her indispensable support and suggestions throughout the internship work.

I would also genuinely like to express my gratitude to the Department Internship Coordinator, Prof.P.P.Joshi, for her constant guidance and support and for the timely resolution of the doubts related to the internship process.

Finally, I would like to thank my mentor, Rahul Kumar for his constant help and support during the overall internship process.

Contents

List of Figures

List of Tables

1 Title

Comparative Analysis of Fusion Models, Convolutional Neural Networks and logical Classifiers for Flame vs Fire Classification"

2 Introduction

Controlled and uncontrolled fires have a significant impact on our environment, economy, and society. Uncontrolled fires can cause severe damage to properties, wildlife habitats, and human lives. On the other hand, controlled fires can cause false alarms in the fire detection system. Therefore, accurately distinguishing between controlled and uncontrolled fires is crucial for effective management and mitigation strategies. In recent years, machine learning techniques, specifically deep learning, have shown great promise in automatic controlled fire and uncontrolled fire classification. Convolutional Neural Networks (CNNs), a type of black box model, have been successfully used for this task, achieving high accuracy and speed. However, these models lack interpretability, which makes it difficult to understand how they arrive at their predictions.

To address this issue, we explore the use of a logical classifier for the classification of controlled vs uncontrolled fires using features derived from the input image. These features have been designed to mimic the way humans analyze images, providing a more intuitive approach to the problem. By combining these human-crafted features with the neural network's learned features, we aim to create a hybrid model that can improve the accuracy of fire classification while maintaining interpretability. In this study, we analyze the performance and interpretability of black box models for controlled vs uncontrolled fire classification and compare it with the performance of the hybrid model that fuses human-crafted features with the neural net.

Despite their interpretability advantages, logical classifiers can face certain challenges in practice. One of the primary challenges is the need for domain expertise and knowledge to create the rules and features that the model relies on. These rules may be complex and require significant effort to develop and refine, which can make the model less accessible to non-experts. Another concern with pure neural net models is that they tend to overfit the training dataset and perform poorly on new data. To overcome these challenges, careful selection and refinement of features, as well as techniques such as cross-validation, regularization, and ensemble methods, may be required. This research aims to improve the accuracy of fire classification and provide more interpretable models, ultimately contributing to the effective management and mitigation of fires.

3 Problem Statement

The problem addressed in this research is the accurate classification of images into flame and fire categories using deep learning techniques. The objective is to develop a fusion model that combines the strengths of two models - a convolutional neural network and a traditional machine learning algorithm - to achieve high accuracy in image classification. The focus is on evaluating the performance of the fusion model and comparing it with standalone models, with the aim of improving the accuracy of flame and fire detection systems.

4 Objectives and Scope

Objectives:

- To develop and compare different image classification models for flame detection using deep learning techniques.
- To investigate the impact of feature fusion techniques on the performance of flame detection models.
- To evaluate the effectiveness of the proposed models based on relevant performance metrics such as accuracy, precision, recall, and F1-score.
- To provide insights into the strengths and limitations of the proposed models and recommend potential avenues for future research.

Scope:

- The study focuses on the development and comparison of deep learning-based image classification models for flame detection.
- The models are trained and tested on a benchmark dataset of images of flames and non-flames.
- The study explores the impact of feature fusion techniques on the performance of the models.
- The evaluation of the proposed models is based on standard performance metrics, including accuracy, precision, recall, and F1-score.

5 Methodological Details

Following is the methodology used to implement the research work. Dataset: We used two datasets of fire images for our experiments: the Flame (Controlled Fire) dataset and the Fire dataset. The Flame dataset contains 482 labeled images of candle flames, and can be found at the following link: `https://github.com/MartinRobomaze/candle-flame-dataset/tree/master/yolo-labels`. The Fire dataset contains 755 labeled images of various types of fires, and can be found at the following link: `https://www.kaggle.com/datasets/phylake1337/fire-dataset`. We resized all the images to have a height and width of 150 pixels for use in training our models. Overall, the datasets provided a diverse range of images that allowed us to explore different types of fires and flames in our experiments.

Masking Script: To accurately extract the fire features from images, it is important to reduce noise and isolate the region of interest. This process involves masking the image to only capture the fire part while minimizing the effect of other sources of light and color in the image.

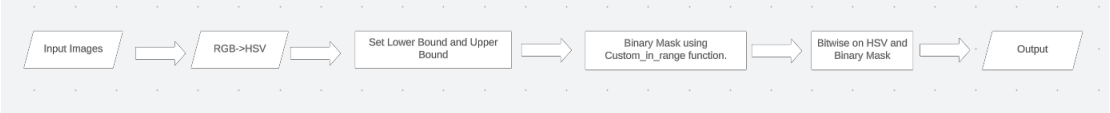


Figure 1: Masking Pipeline

The procedure for the script was as follows:

1. Convert the input image to the HSB color space.
2. Compute the 75th percentile value of the B channel and assign it to a variable X.
3. Define the lower and upper bounds for each value of the HSB color space as follows:

   ```
   Lower bound = [0, 70, X]
   Upper bound = [35, 255, 255]
   ```

4. Create a binary mask for the input image using a custom inRange function that checks if the Hue, Saturation, and Brightness of each pixel falls within the established bounds. Set the mask value to 1 for pixels that meet this condition, and 0 for those that don't.
5. Apply a bitwise AND operation between the binary mask and the HSB image to obtain an output image that captures only the fire features while minimizing the effects of other sources of light and color in the image.

The custom_inRange function was implemented to address the limitations of the standard OpenCV inRange function in capturing the whites inside the fire. The custom_inRange function utilizes the lower and upper bounds for each value of the HSB channel established in the previous step, as follows: Mask[i][j] = lower_bound_hue ¡ Hue ¡ upper_bound_hue and lower_bound_saturation ¡ Saturation ¡ upper_bound_saturation and lower_bound_value ¡ Value ¡ upper_bound_value Additionally, an extra condition is added to preserve the whites inside the fire,

whereby if the Value of the pixel is greater than or equal to 250 and the Hue is less than or equal to 60, the mask value is set to 1. This ensures that the binary mask captures all the necessary features of the fire, including the whites inside it, to achieve accurate feature extraction. For the lower bound's B value, we find the value of the 75th percentile of all B values in the image. This is to ensure that features from images with differing brightness are extracted properly. It also helps capture low intensity which the model may miss.

Logical Features Constructed:

Vertical Intensity Arrays (VIA) We developed a novel approach called the Vertical Intensity Arrays (VIA) method to classify images as controlled fire or uncontrolled fire. The VIA method was inspired by how humans visually categorize fires based on factors such as size, spread, and color. Our goal was to create a mathematical representation of the visual characteristics of fire. For each image in our dataset the steps taken to calculate VIA were 1. Generated an array of shape 1 x n, where n is the horizontal length of the image where each value in the array is the sum of the number of orange and yellow pixels in the image for that position along the horizontal axis. 2. For every x-coordinate we summed the total number of pixels with yellow or orange HSV values. For example, if the shape of the image is 5 x 5, its VIA will be of size 1 x 5. Finally, to visualize the intensity profile of the fire, we plotted the VIA with x-coordinates of the image on the x-axis and pixel counts on the y-axis. We observed that the plot had a shape similar to that of fire. Standard Deviation-Spike-Fall (SSF) In addition to the Vertical Intensity Array (VIA), we developed a simplified version called the SD-Spike-Fall (SSF) array to further classify images as controlled or uncontrolled fire. The SSF array captures three values: 1. The standard deviation of the VIA 2. The number of spikes (abrupt rises) in the VIA 3. The number of falls (abrupt drops) in the VIA.

Total Arc Length (TAL) Another approach to classifying controlled and uncontrolled Fire is by differentiating by the number of contours they produce. Usually, uncontrolled Fires are made up of many smaller individual fire and smoke particles which in turn increases the number of contours. By adding up the perimeters of all contours we get the Total Arclength.

Logical Classifier: In this study, a threshold-based logical classifier for detecting controlled and uncontrolled fire has been developed. The classifier employs a set of rules based on statistical parameters. The primary criterion for detection is the standard deviation of the VIA, which must be less than 2500. If this condition is not met, the output of the classifier is set to "fire detected". If the standard deviation is less than 2500, the presence or absence of spikes and falls in the VIA is checked. This acts as a primary condition for detection of uncontrolled fire. If the number of spikes and falls are both equal to zero, then the image is classified as a fire. Otherwise, the difference in the number of spikes and falls is calculated, and if it exceeds twice the minimum of the two, the classifier checks the Arc length parameter generated for that image. If it is less than 642, the output is set to "Controlled Fire detected". Otherwise, the output is set to "Uncontrolled Fire detected". If the difference in the number of spikes and falls does not exceed twice the minimum of the two, the Arc length parameter is checked, and if it is greater than 1883, the output is set to "Uncontrolled Fire detected". Otherwise, the output is set to "Controlled Fire detected". The classifier has been tested on the dataset, and the results have shown a good amount of accuracy in classification.

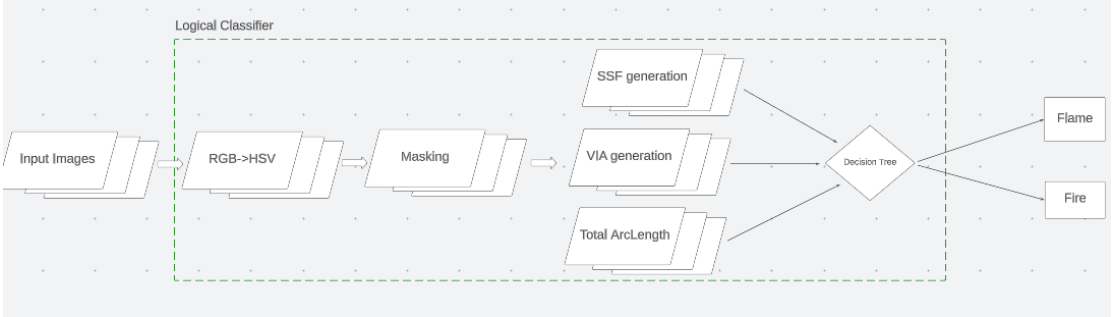


Figure 2: Logical Classifier Diagram

BlackBox Model As the Black box model, we have trained and tested two convolutional neural nets on the same dataset to find out the classification accuracy of each. The first neural net consists of two convolutional layers, each with 64 filters of size (3,3) and ReLU activation, followed by a max-pooling layer with pool size (2,2). The purpose of the convolutional layers is to extract meaningful features from the input images, while the max-pooling layers reduce the spatial dimensionality of the features and help prevent overfitting. After the convolutional layers, a flattened layer is added to convert the 2D feature maps into a 1D feature vector. This vector is then passed through a fully connected layer with 128 units and ReLU activation, which acts as a classifier on top of the extracted features. Finally, a dense layer with 2 units and SoftMax activation is used to produce the final classification probabilities for the two classes. During training, the model utilizes the Adam optimizer and sparse categorical cross-entropy loss function, while the accuracy metric is employed for evaluation purposes. The dataset used for training and validation is split into 80:20 ratio, with 80Following is the summary model of the neural net.

Model	Layer (type)	Output Shape	Param
Fusion Model	input_1 (Input Layer)	(None, 150, 150, 3)	0
	conv2d (Conv2D)	(None, 148, 148, 64)	1792
	max_pooling2d (MaxPooling2D)	(None, 74, 74, 64)	0
	conv2d_1 (Conv2D)	(None, 72, 72, 64)	36928
	max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
	flatten (Flatten)	(None, 82944)	0
	input_2 (InputLayer)	(None, 3)	0
	dense (Dense)	(None, 128)	10616960
	dense_1 (Dense)	(None, 128)	512
	concatenate (Concatenate)	(None, 256)	0
	dense_2 (Dense)	(None, 128)	32896
	dense_3 (Dense)	(None, 2)	258
Sequential Model	Conv2D (Conv2D)	(None,148,148,64)	1792
	MaxPooling2D (MaxPooling2D)	(None, 74, 74, 64)	0
	Conv2D_1 (Conv2D)	(None, 72, 72, 64)	36928
	MaxPooling2D_1 (MaxPooling2D)	(None, 36, 36, 64)	0
	Flatten (Flatten)	(None, 82944)	0
	Dense (Dense)	(None, 128)	10616960
	Dense_1 (Dense)	(None, 2)	258

Table 1: Model Architectures

Fusion /Impure Model The impure model is a fusion of traditional convolutional

neural networks with external features supplied by us. For each logical feature viz. SSF, VIA and Total Arc Length the 64x2 neural network is fed with an external feature set representing the logical feature and hence a merged model is formed. First, we define the architecture of the image model. It has an input layer that takes images of shape (150,150,3), where 3 corresponds to the RGB channels of each pixel in the image. This input layer is followed by two convolutional layers with 64 filters of size (3,3) each, and ReLU activation function. A max-pooling layer is then applied with pool size (2,2) to down sample the feature maps. The same pattern is repeated with another set of convolutional and max-pooling layers, followed by a Flatten layer that converts the output tensor into a 1D tensor, which is then passed to the next layer. Next, we define the architecture of the secondary input model. It has an input layer that takes in a 3-dimensional tensor with shape (3,), which will be used to pass in some additional information related to the images. This input layer is followed by a dense layer with 128 units and ReLU activation function.

The output of the image model and the secondary input model are then concatenated together using the concatenate layer. This merged output is then passed through two dense layers, each with 128 units and a ReLU activation function. The output is passed through a dense layer with 2 units and SoftMax activation function, which gives us the probability of the image belonging to each of the two classes.

Finally, we compile the model using the Adam optimizer, and a sparse categorical cross-entropy loss function as we have integer labels, and accuracy as the metric. The model is then trained using the fit method by providing both the image data and secondary input data as inputs, along with the corresponding labels. Following is the summary table of the Fusion Model.

6 Modern engineering tools used

In this research project, we used a range of modern engineering tools to develop and evaluate our methods for detecting and analyzing fires in images and videos. We utilized popular programming languages such as Python and MATLAB to develop the computer vision algorithms that were used for image and video processing. In addition, we used various Python libraries such as OpenCV, NumPy, and Pandas to facilitate image manipulation and data analysis.

To train and evaluate our machine learning models, we used popular frameworks such as Keras and TensorFlow, which provided a high-level interface for building and training deep learning models. These tools allowed us to experiment with different architectures and hyperparameters to find the best models for our task.

Finally, to visualize our results, we used tools such as Matplotlib and Seaborn to create informative plots and charts. These tools allowed us to quickly visualize the performance of our models and gain insights into the characteristics of the fire images and videos. Overall, the use of these modern engineering tools greatly facilitated the development of our methods and helped us to achieve state-of-the-art performance in fire detection and analysis.

7 Outcome/ results of internship work

The performance of all the models was compared based on F1 score as there was an imbalance between the dataset size of Uncontrolled Fire and Controlled Fire. The following table summarizes the F1 score As seen in the performance table,

Model Name	**F1 Score (performance)**
64*2 Neural Net (unmasked)	0.91
64*2 Neural Net (masked)	0.97
128*3 Neural Net (unmasked)	0.91
Fusion Model (SSF,64*2)	0.76
Fusion Model (VIA,64*2)	0.51
Fusion Model (TAL,64*2)	0.95
Pure Logical Classifier	0.79
Stand-alone Logical Feature (TAL)	0.68
Stand-alone Logical Feature (SSF)	0.75

Table 2: Performance of different models

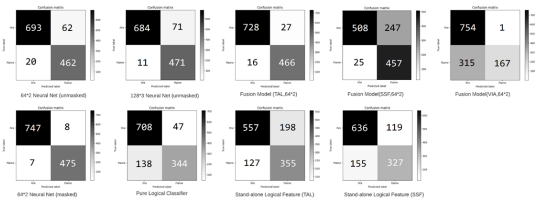


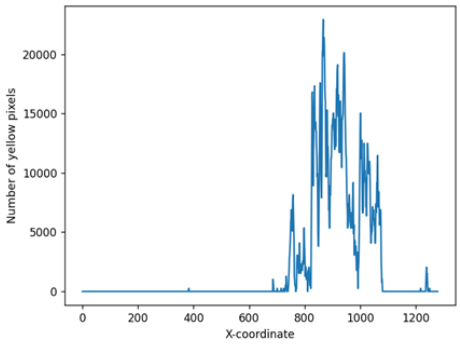
Figure 3: Confusion Matrices for all the models

the neural net trained on masked images performed with highest f1 score (0.97) followed by the Fusion Model with Total Arc Length as the concatenated logical feature at 0.95. This shows that by masking, that is, on removing the noisy background, the neural net adapts to extract more meaningful features to classifying an image as controlled or uncontrolled fire. Being a black box, it is hard to point out which exact features are at play to achieve that performance. But by analyzing a small set of test images containing standard, noisy and outlier images we speculate that the features extracted by the neural net were related to containment of the fire base and the uniformity of flame/intensity spread throughout the image. Overall, the Neural nets performed much better than stand-alone logical features, but when combined to form Fusion Models, their performance shows improvement. The key finding here was that even when trained on unmasked images the Fusion Model (TAL) was at par with the neural net trained on masked images. This suggests that the provided Total Arc length feature was an excellent logical baseline for the model when it encountered outliers while classifying. The Fusion Models constructed with other logical features like SSF and VIA did not fuse well with the neural net, and hence can be seen as nothing more than noisy classifiers.

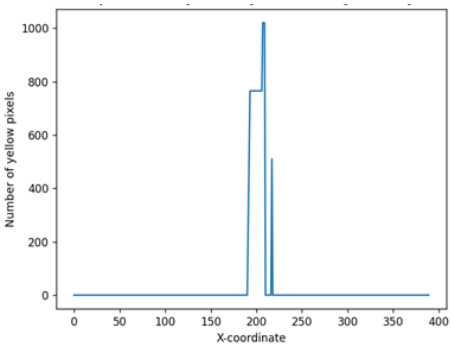
The results and observations for individual logical features are as follows: VIA: By plotting the VIA with x-coordinates of the image on the x-axis and pixel counts on the y-axis, we observed that the plot had a shape similar to that of fire. We compared the VIA of several images of uncontrolled and controlled fire and calculated the standard deviation of the pixel counts. Our comparative study revealed that images with perceptible uncontrolled fire had a greater deviation in

(a) Figure 2.1 Uncontrolled Fire

(b) Figure 2.2 Controlled Fire (Flame)



(c) Figure 2.3 VIA



(d) Figure 2.4 VIA

Figure 4: Input Images and their Vertical Intensity Arrays Plot

pixel values compared to those with controlled fire. Additionally, we observed that images with controlled fire had a steep increase and decrease in pixel counts instead of a tapered one. SSF: Through observation, we found that images containing Uncontrolled fire typically had a higher standard deviation and either no spikes or falls, or a vague number of spikes and falls. In contrast, images with controlled fires tended to have a lower standard deviation and an equal number of spikes and falls. The equal number of spikes and falls was due to the symmetric shape of the controlled fires. As seen in Figure 2.2 The spike, fall count, and standard deviation were 2,2 and 170.3 respectively whereas Figure 2.1 had its spike, fall count, and standard deviation 0,0 and 4600.4 respectively. TAL: Through our observation, we found that images containing uncontrolled Fire mostly had a higher Total Arclength than Controlled Fire. Total Arc Length is highly dependent on image resolution. Higher image resolutions increase Total Arc Length classifying power. Figure 2.1, and Figure 2.2 have TAL values 3987.62, 680.12 respectively.

The accuracy of the individual logical features depends heavily on selecting the right threshold value which will be the right fit for a large dataset, proper threshold value can be found out either by estimation or brute force techniques which will again basically imitate the working of a neural net.

8 Achievements and Publication

This research project has been a valuable learning experience for us. We have gained in-depth knowledge of image processing techniques, machine learning algorithms, and their applications in fire detection. We learned about the fundamental concepts behind image processing, such as image filtering, feature extraction, and segmentation. We also learned about different machine learning algorithms, such as neural networks, support vector machines, and decision trees, and their suitability for different types of data. Furthermore, we learned about the importance of data preprocessing, feature selection, and hyperparameter tuning in developing accurate and robust machine learning models.

In addition to technical skills, this project has also helped us develop soft skills, such as teamwork, communication, and time management. We worked collaboratively as a team, shared our ideas and knowledge, and helped each other overcome challenges. We also learned how to communicate our research findings effectively, both in writing and in oral presentations. Overall, this research project has provided us with a valuable opportunity to apply our theoretical knowledge to real-world problems, learn new skills, and develop both technical and soft skills that will be beneficial for our future careers.

Publishing this research work is one of our primary objectives, as it will allow us to disseminate our findings and contributions to the wider scientific community. We believe that our work will be particularly valuable to researchers and practitioners in the fields of computer vision and fire detection, as it provides a new approach to fire detection that can help to improve safety and prevent damage to property. We plan to submit our work to top-tier academic conferences and journals in the field, including IEEE Transactions on Image Processing and CVPR, to ensure that it is rigorously reviewed and reaches the widest possible audience. In addition, we also plan to publish our code and data sets on open-access platforms such as GitHub and arXiv, so that other researchers can replicate and build upon our work.

Overall, we are excited about the potential impact that this research project can have, and we are committed to sharing our findings and contributions with the broader scientific community. We believe that our work has the potential to make a significant contribution to the field of computer vision and fire detection, and we are confident that our approach will be useful for other researchers and practitioners working in this area. Ultimately, we hope that our work will contribute to improved safety and security in a range of settings, including industrial facilities, commercial buildings, and residential homes, and we look forward to sharing our progress and results with the wider scientific community.