COMPUTER VISION REPORT

FACIAL EMOTION RECONGITION

IMAGE CLASSIFICATION

Folder Link: https://drive.google.com/drive/folders/1IjJvU2nmUTkKCspRyAlcuwZHaQTwGy8t?usp=sharing

I. INTORUDCTION

The report intends to highlight the project aim of implementing a pipeline for classifying facial emotion recognition by using different combinations of feature extractors and different machine learning algorithms. Our aim is to look at two different feature extractors, the first being scale-invariant feature transform (SIFT) and the second being histogram of oriented gradients (HOG). These will be paired with two of our classification algorithms which are support vector machine (SVM) and Multi-Layer Perceptron (MLP), as for the third classification algorithm, we will be looking at a convolutional neural network (CNN). We will also apply our best model to a short video and see if we can detect faces and classify facial emotions. The report structure consists of data, methods which is split into feature extractors and classification, were we will look at these two sections in more detail alongside the pipeline, the report will also include a look at the results and a conclusion.

II. DATA

The data we will be using is called Real-world Affective Faces (RAF) Database [1], which consists of JPG images obtained from the Internet. The data is split into a training set of 12271 images and a testing set of 3068 images. All images have been roughly aligned using similar transformations according to the two eye locations and the centre of mouth and then resized to 100*100. The labels are classified as 7 different types of emotions: 1: Surprise, 2: Fear, 3: Disgust, 4: Happiness, 5: Sadness, 6: Anger and 7: Neutral.
We will also be using a short tiktok video which was downloaded from youtube and converted into avi format.

III. METHODS

In this section we discuss the key concepts we will use to build our facial recognition function.

A. Feature Extractors

Feature extraction is one of the core steps in image recognition. It is the process of deriving useful features that define objects in an image. The features are informative and non-redundant and are function of the measurement variable [2]. These features help the improve the effectiveness of the classification. There are several techniques but for this project we will concentrate on SIFT and HOG.

1) SIFT

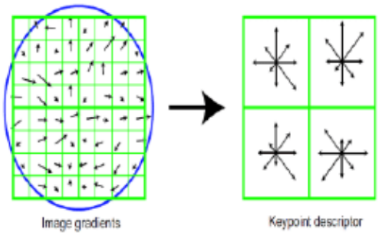
The SIFT algorithm which is known formally as scale-invariant feature transform is a feature extractor that was first introduced by Lowe in 1999. SIFT algorithms extracts points of interest on an image known as high contrast regions such as edges and corners, this allows the features to be invariant to image translation, scaling, rotation or even illumination. The algorithm is divided into two modules which are key point detection and descriptor generation.

The algorithm is split into 4 steps [2]:

Scale-space extrema detection – key points in the SIFT framework are detected. A Gaussian difference filter is applied, and the local minimum and maximum are found.

Key-point localization – The key points are chosen with high contrast points. The Taylor expansion of DOG (difference of gaussian) scale-space function and the candidate key point is taken as the origin.

Orientation assignment – Orientation is allocated for key points constructed on local image gradients; this leads to the property of rotation invariance.



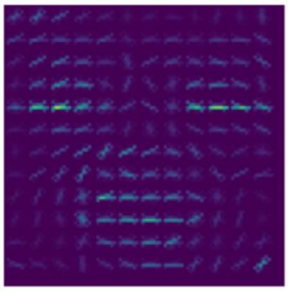
Key point descriptor – Descriptor vectors are computed for each key point. The descriptors are partially invariant to illumination and are distinctive.

2) HOG

The HOG algorithm which is known formally as histogram of orientated gradients is feature extractor first described in 1986 by R. K. McConnell but was later supplemented by Dalal in 2005. HOG counts the occurrence of gradient orientation in small areas of an image which is computed on a dense grid of uniformly spaced cells. All the information obtained is collected into a single vector [4].

The algorithm can be summarised in 3 steps [5]:

Gradient computation – Gradient magnitude and angels are computed using the computed vertical and horizontal spatial gradients.



Orientation Binning – Cells are created by dividing the image into small, connected regions. Depending on the gradient angle each gradient magnitude of each pixel in placed in origination bins.

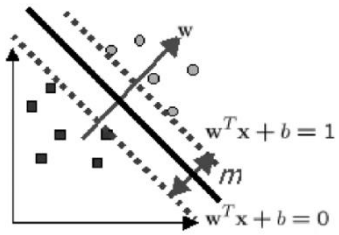
Feature description – Blocks are created by grouping adjacent cells, normalisation is then applied to each block. A descriptor is then created by the concatenation of the normalised block histograms in a detection window.

B. Classification

The next step in the pipeline is to select the algorithm to classify the feature descriptors to start training the model.

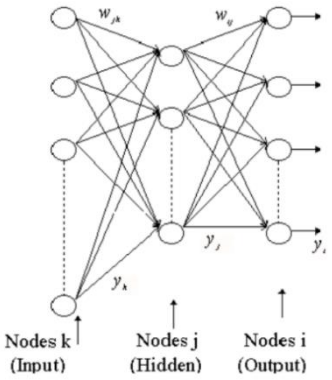
1) SVM

The SVM is a classification algorithm capable of preforming both linear and non-linear classification, this is achieved using decision boundaries known as hyperplanes and maximising the distance between the two boundaries [6]. Instances called support vectors determine the decision boundaries. The algorithm is setup for binary classification but can be adapted for multi-class using one-vs-one and one-vs-all methods. Adjusting the width of the hyperplane so all instances are on the right side of the decision boundary is called a hard margin classification, this is true if the data is linear separable but may be sensitive to outliers. The algorithm aims to balance between limiting the number of violations and increasing the width of the hyperplane. A soft margin classification can occur when violations are found.
SVM can address nonlinear classification using polynomial features, a kernel trick is applied to avoid the huge number of features because of a high polynomial degree, the trick moves the data into a 3-D space and designs a decision surface to differentiate between the classes, the data is then returned to its original form which allows for the same results but without adding any new features.



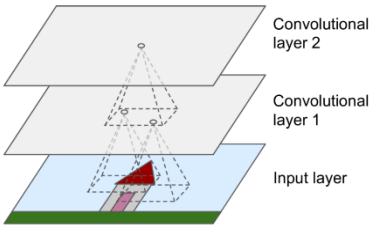
2) MLP

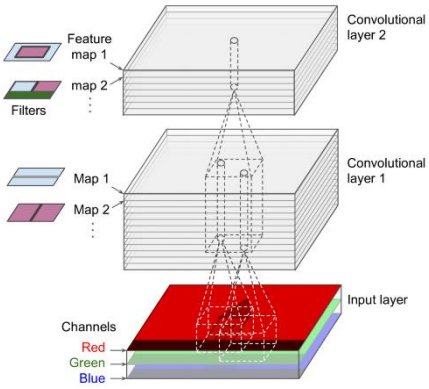
In the late 50s Frank Rosenblatt introduced the simple artificial neural network called the perceptron. Its fundamentals are based on artificial neurons called threshold logic unit (TLU) or linear threshold unit (LTU), these connected neurons have weights associated with them. The weighted sum of all the inputs is computed by the TLU and applies the step function to produce the results. MLP is defined as an extension of this by having one or more layers of TLUs called hidden layers with an input and out layer. Every layer has a bias layer except the output layer and all the neurons are fully connected. A deep neural network (DNN) is defined as a MLP with two or more hidden layers. Backpropagation training algorithm was introduced by Rumelhart in the mid-80s which used gradient decent optimisation. It consists of an iterative process of forward pass where the output of each neuron is calculated for each layer with respect to the target variable. After calculating the error, the algorithm calculates its contributions by stepping through the network backwards. Gradient decent is the final step in the algorithm where the weights are adjusted to reduce the overall weight.



3) CNN

Convolutional neural networks are based on the study of brain's visual cortex and have been used in image recognition since the 80s. DNNs work great for small images but break down for larger images hence the use of CNNs. The standout building block is the convolutional layer. Neurons in the first convolutional layer are not connected to every pixel in the input image but instead only to the respective fields. The Second convolutional layer is connected only to the neurons in the respective field of the first convolutional layer. This architecture concentrates on low level features in the first layer and assembles them to into a larger higher-level feature in the next hidden layer. Hierarchical structures are common in images hence why CNN works well for image recognition [7].



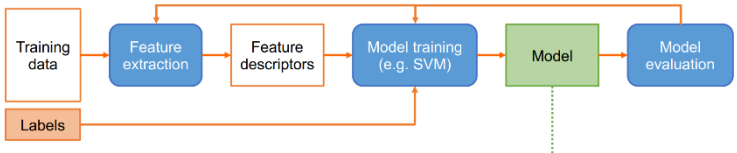


The weights in a CNN are the dot product produced with each pixel to produce a new pixel, resulting in a small matrix acting as an image filter. The produced output by each neuron is combined for each layer and passed as the inputs to every neuron in the next layer until the end. Each convolutional layer has multiple filters and outputs a feature map for each filter. In addition, they also have a pooling layer which has a goal of subsampling the input images to reduce the computational load, memory and number of parameters.

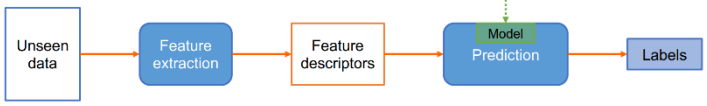
IV. IMPLIMENTATION

Before we start, we must first adopt a pipeline in which we will follow to help us achieve our aim, below is the pipeline we will adopt:

- Training



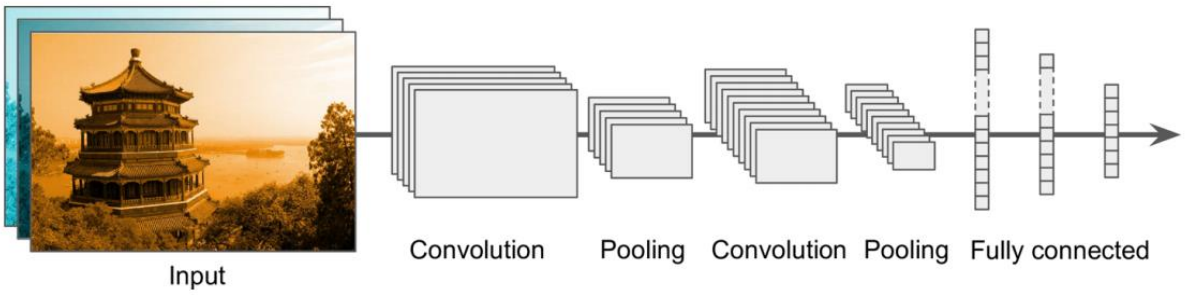
- Testing/prediction/inference



The dataset is already split between training and test data. In terms of validation, traditionally we would opt for K-fold validation but due to computational power we will adopt the introduction of a validation set. As a result, we will be further splitting the training data at an 80%:20% ratio to create the validation set, this ratio is adequate given our dataset is a large one. Usually, we would also adopt a grid search to optimise the hyperparameters but due to computational power we will adopt a manual random grid search.

In terms of the process, for each Feature extractor SIFT & HOG we will run and train a SVM and MLP model. The combination with the highest accuracy tested on the validation set will be the model to further optimise its hyperparameters and finalise as one of our three models. As for the CNN model, no need to pair it with a feature extraction algorithm as its convolutional layers act as feature extractors.

In terms of metrics, we will be comparing all three models based on their accuracy level. In terms of optimising, we will look to tweak the hyperparameters and test on the validation set. For SVM model we will look at the hyperparameters C - misclassification cost, gamma, degree and kernel. For MLP we will be looking at number of neurons, learning rate and optimiser. As for CNN we will be adopting the foundational structure of a LeNet CNN but will make amendments to it to make it our own and then we will focus on the following parameters momentum, learning rate and optimiser.



For our final testing function, we will bring in the test data, apply feature extraction and apply our three final saved models. For the video section we will adopt a similar approach where we will bring in the video and detect faces in the video using face cascade function from cv2 library. The next step will be to isolate the faces, resize them and finally load and apply our best model only.

V. RESULTS

After the initial results were collected, we can see that our best models for each combination is SIFT-SVM at 40% accuracy, HOG-MLP at 70% accuracy and CNN at 20% accuracy, we then further developed these models to improve their accuracies.

Initial Accuracy		
Feature	Model	Accuracy
HOG	SVM	69%
HOG	MLP	70%
SIFT	SVM	40%
SIFT	MLP	39%
-	CNN	20%

For HOG-MLP, our best performing model achieved an accuracy of 72%. After hyperparameter tunning our final model consisted of 25 neurons, learning rate of 0.0001 and Adam optimiser. We only managed a 2% accuracy increase from our initial results.

For SIFT-SVM, our best performing model achieved an accuracy of 42%. After hyperparameter tunning our final model consisted of C equal to 5, degree equal to 1 and gamma equal to 7. We only managed a 2% accuracy increase from our initial results.

For CNN, our best performing model achieved an accuracy of 62%. After hyperparameter tunning our final model consisted of momentum equal to 0.8, learning rate of 0.0001 and Adam as an optimiser. We were able to improve the model accuracy by 42%.

When we use our models on the test data they perform pretty much the same except the CNN model which produced an accuracy of around 10%, this means our model was overfitting and doesn't do really well on generalising on new data.

SVM

Hyperparameter		C		
Value		0.5	5	10
Kernel	Linear	39%	40%	39%

Hyperparameter		Degree		
Value		1	3	7
Kernel	Poly	40%	39%	40%

Hyperparameter		Gamma		
Value		1	5	7
Kernel	RBF	39%	41%	42%

MLP

Hyperparameter	Number of Neurons		
Value	10	25	50
	64%	72%	71%

Hyperparameter	Learning Rate		
Value	0.01	0.001	0.0001
	60%	64%	72%

Hyperparameter	Optimizer		
Value	ADAM	SGD	LBFGS
	72%	60%	69%

CNN

Hyperparameter	Momentum		
Value	0.6	0.8	1
	57%	62%	30%

Hyperparameter	Learning Rate		
Value	0.01	0.001	0.0001
	49%	58%	62%

Hyperparameter	Optimizer		
Value	ADAM	SGD	ADAGRAD
	14%	62%	13%

In terms of using our best model to test on a video, the cascade face detection did manage to detect some faces. It also managed to detect other parts of the video that were not faces, that resulted in random images in our set. In terms of the prediction, it seemed to predict correct as the face is somewhat close to the prediction label, as see on the right which predicted a surprised emotion.

VI. CONCLUSION

The project aim was to classify facial emotion recognition by using different combinations of feature extractors and different machine learning algorithms. We used SIFT and HOG as feature extractors and SVM, MLP and CNN as classifiers. In conclusion our overall the best feature extractor was HOG which achieved an accuracy of around 70% for each of its classifiers. Our overall best model was HOG-MLP achieving an accuracy of 72%, this was unexpected given the fact we also used a CNN is designed for image recognition with its convolutional layers, filters, and pooling layers. WE could have further improved all models in reality if we had better computational power to better optimise the models using a full grid search instead of a manual random grid search which is very limited.
To further improve the project, potentially running our best model on the video displaying the face detection and predicted label as it plays.

VII. REFERENCES

[1] Li, S., Deng, W. and JunPing, D., (2017). Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pp 2584—2593.

[2] Yuvaraju, M., Sheela, K. and Sobana Rani, S., (2015). Feature Extraction of Real-Time Image Using Sift Algorithm. International Journal of Research in Electrical & Electronics Engineering. 3. 1-7.

[3] Lowe, D.G., 2004. Distinctive image features from scaleinvariant keypoints. International journal of computer vision, 60(2), pp.91-110.

[4] Haythem, Ameur & Helali, Abdelhamid & Nasri, Mohsen & Maaref, H. & Youssef, Anis. (2014). Improved feature extraction method based on Histogram of Oriented Gradients for pedestrian detection. GSCIT 2014 - Global Summit on Computer and Information Technology. 1-5. 10.1109/GSCIT.2014.6970120.

[5] Zhou, W., Shengyu, G., Zhang, L. and Lou, X., (2020). Histogram of Oriented Gradients Feature Extraction from Raw Bayer Pattern Images. IEEE Transactions on Circuits and Systems II: Express Briefs. PP. 1-1. 10.1109/TCSII.2020.2980557.

[6] Frias-Martinez, E., Sanchez, A. and Velez, J., (2006). Support vector machines versus multi-layer perceptrons for efficient off-line signature recognition. Engineering Applications of Artificial Intelligence, 19(6), pp.693–704.

[7] Aurelien, G., (2017). Hands-On Machine Learning with Scikit-Learn & TensorFlow. O'REILLY, (pp.361-386)