IN3060/INM460 Computer Vision Coursework report

- **Student name, ID and cohort:** Ethan Chew (180037454) - PG
- **Google Drive folder:** https://drive.google.com/drive/folders/1tSklg2KJJCNM2U2tI0pSZUjvQ-84u5cQ?usp=sharing

Data

Labelled Images Dataset

- The dataset is made up of a training and testing set, with 12,271 and 3,068 images respectively.
- Images in both sets are of the same dimensions, specifically: (100, 100, 3). Essentially the images are made up of 3 colour channels (RGB) with an equal height and width of 100 pixels. An example is shown in Figure 1.
- With both, the labels are stored in their respective text files but are matched through their filenames when loading and importing the dataset.

Figure 1: Example Image

- With each image, an associated emotion label, out of the 7 possible emotion labels, is provided. In particular, the emotion labels include: surprise, fear, disgust, happiness, sadness, anger and neutral, and are tagged by class labels from 1 to 7 respectively.
- The distribution of the training set is fairly imbalanced with most labels being happy, sad and neutral. Figure 2 shows the distribution of classes across the training set.

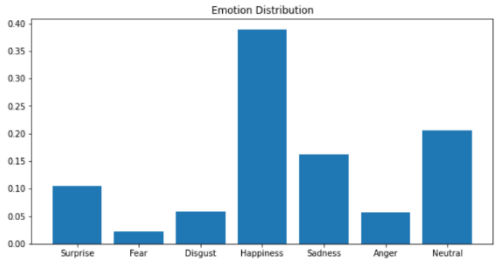

Figure 2: Distribution of Classes over Training Set

Video Dataset

- The video used is a self-recorded one which captures the 7 emotions and has dimensions of (113, 1080, 1920, 3). Essentially, the video is made up of 113 frames and each frame is a coloured image (3 colour channels – RGB) with height and width of 1080 and 1920 pixels respectively. A single frame is shown in Figure 3 as an example.
- Note that the original video was sped up 8 times (hence the few number of frames) to work around memory limitations.

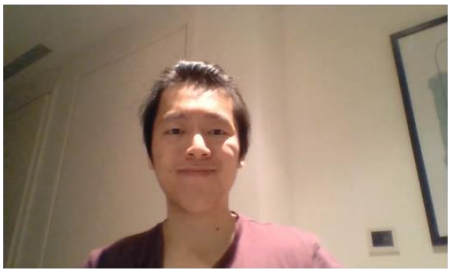

Figure 3: 60th Frame in the Video (Happiness)

Implemented methods

Overview

- In total, 3 models are trained, specifically: Multilayer Perceptron (MLP), Support Vector Machine (SVM) and Convolutional Neural Network (CNN).

MLP and SVM

- Instead of the raw image intensity values, the MLP and SVM models are trained over a set of feature descriptors – Histogram of Oriented Gradients (HOG). To summarise, HOG first converts the image to grayscale and divides it into blocks and then cells. In each cell, the gradient magnitude and angles are computed and the angles are subsequently tallied over a histogram weighted by its magnitude. Here we binned the gradient angles into 8 bins and utilised a cell of size (8, 8).
- Other feature descriptor techniques exist, like Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF), but HOG was used over both models to allow for a fair comparison.
- Hyperparameter tuning and grid-search is adopted for both models, allowing for comparison of model performance over different hyperparameter combinations. For MLP, the size of the hidden layer, learning rate and batch size are varied. For SVM, the choice of kernel, polynomial order (for polynomial kernel) and size of regularisation parameter are varied instead.
- Additionally, k-fold cross validation is adopted when training over each hyperparameter combination. In particular, 10 folds are used whereby the training set is divided into 10 partitions and for each fold, 9 partitions are used for training while the remaining is left for validation.

- The trained model over each fold and hyperparameter combination is saved and for each hyperparameter combination, the model with the highest f-score is selected over the 10 folds (i.e. 10 models down to 1 model for each hyperparameter combination). The f-score is used since it quantifies the trade-off between precision and recall (i.e. harmonic mean of precision and recall).
- For each model type, the best performing model (and hyperparameter combination) is selected and is one which ideally has: a high validation f-score, precision and recall, and short training times.

CNN

- When training a CNN, the images are first normalised and parsed through a series of convolutional and pooling layers and the resulting feature map is flattened and passed through a fully connected neural network for image classification.
- Due to the computational intensity, a set of CNNs are not trained using k-fold cross validation. Instead, the training set is split into training and validation using a 90-10 split respectively.
- Similar with training our MLP and SVM model, hyperparameter tuning and grid-search is adopted. In particular, kernel size in our convolutional layer and learning rate in our weight update rule is varied. With each combination, 3 convolutional layers and a fully connected neural network with hidden layers of sizes 120 and 84 are utilised. Additionally, the loss is computed via Cross Entropy Loss and the network is optimised via Stochastic Gradient Descent. The model selection process to determine the best performing model follows what was seen with MLP and SVM.

Facial Emotion Recognition Video

- For every 10 frames, we utilise a pre-trained face detection model (following the Viola-Jones approach) for facial detection. Essentially, a set of Haar wavelets are applied over a sliding window to identify frontal faces in an image (or a single frame).
- In each of those frames, the corresponding coloured cut-out of the identified face is then rescaled to a height and width of 100 and a set of feature descriptors are obtained using HOG (we will see that SVM is our best performing model). This is done because our SVM was trained over HOG feature descriptors, obtained from images of size (100, 100, 3).
- The feature descriptors are then fed to our SVM to obtain a predicted emotion.

Results

Model Training

- As mentioned above, the hyperparameters of the 3 models are tuned using grid-search and the results are shown in Figures 4, 5 and 6 accordingly. In each figure, the validation precision, recall and f-score are displayed, along with the time taken to train each model. Additionally, the best performing model for each respective model type is highlighted in green.

Hidden Layer	Learning Rate	Batch Size	Precision (Validation)	Recall (Validation)	F1-Score (Validation)	Training Time
(100, 100)	0.001	200	0.652308	0.653627	0.650865	218.374145
(100, 100)	0.001	300	0.671408	0.677524	0.668328	201.007436
(100, 100)	0.010	200	0.654328	0.655257	0.654166	125.451571
(100, 100)	0.010	300	0.643841	0.643032	0.643076	183.110960
(200, 200)	0.001	200	0.653217	0.651997	0.651481	414.437308
(200, 200)	0.001	300	0.662188	0.673187	0.666101	363.609228
(200, 200)	0.010	200	0.662956	0.665852	0.664204	229.125739
(200, 200)	0.010	300	0.675879	0.677262	0.676262	246.803585

Figure 5: Grid-Search MLP

Kernel	Polynomial Order	Regularisation	Precision (Validation)	Recall (Validation)	F1-Score (Validation)	Training Time
linear	None	0.5	0.643241	0.644951	0.643530	86.913573
linear	None	0.1	0.677389	0.685412	0.678498	75.998388
poly	2	0.1	0.698286	0.700896	0.681379	111.936779
poly	2	0.5	0.691024	0.697637	0.690267	80.522626
poly	5	0.1	0.695397	0.705786	0.696919	124.584914
poly	5	0.5	0.700770	0.705786	0.700827	116.106396
rbf	None	0.1	0.541036	0.601467	0.513254	187.151584
rbf	None	0.5	0.717230	0.701711	0.676480	128.679267

Figure 4: Grid-Search SVM

Kernel Size	Learning Rate	Precision (Validation)	Recall (Validation)	F1-Score (Validation)	Training Time
5	0.001	0.656003	0.667752	0.658879	1304.131916
5	0.010	0.652864	0.660423	0.655498	1299.249579
7	0.001	0.633053	0.638436	0.634400	1684.904034
7	0.010	0.645828	0.655537	0.648706	1683.088641

Figure 6: Grid-Search CNN

Model Testing

- The best performing model of each model type are then tested over the testing set and the performances are evaluated through confusion matrices as shown in Figure 7.

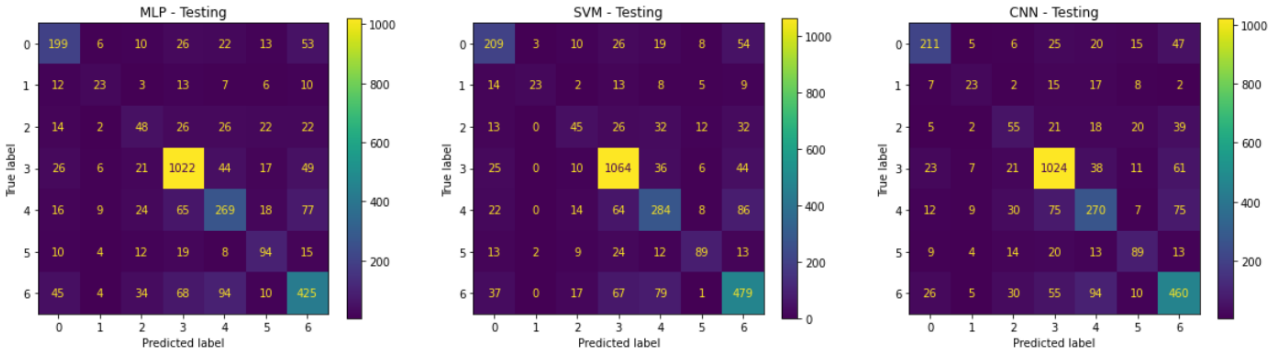

Figure 8: Confusion Matrices of Each Model

- Figure 8 shows the predicted emotion from each model over a sample of 4 testing images.

Figure 9: Comparison of Model Prediction Over Sample of Images

- Between the 3 candidate models, the best performing model (SVM in our case) is applied onto our video and Figure 9 displays its performance over 3 random video frames. The red block captures the detected face and the corresponding emotion prediction is shown to the right of it.

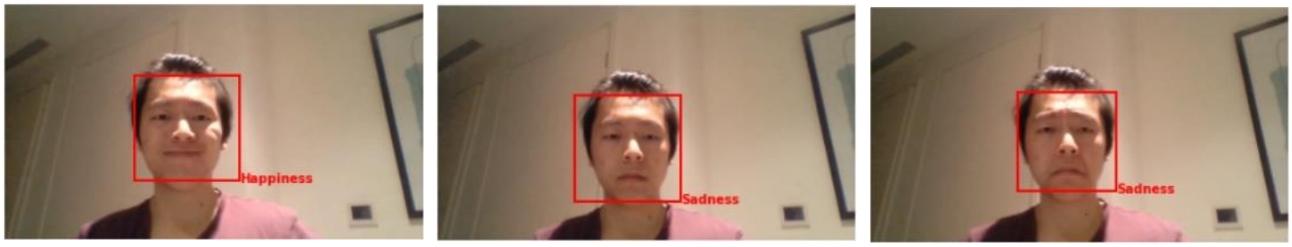

Figure 10: Model Prediction Over Sample of Video Frames

Discussion

MLP

- From Figure 4, we generally see a MLP's performance to improve with larger hidden layers, learning rates and batch size. However, training times also increases with larger hidden layers and batch size.
- More nodes allow for better approximation (though overfitting should be kept in mind if too many layers and/or nodes are employed) but increases the number of parameters to be estimated and increases the computational intensity during backpropagation.
- Smaller batches allow for faster training times since the gradients can be evaluated more quickly but might cause for instability since the gradient will only depend on a single sample.

- Learning rate dictates the speed of parameter updates and higher learning rates may lead to faster optimisation but run the risk of inaccurate and sporadic updates resulting in non-optimal minimas. However, the higher learning rate still governed accurate updates whilst improving training times.
- The selected model showed the best model performance across the 3 measures and decent training times as well with an f-score of 67.62% and training time of 246.80 seconds.

SVM

- From Figure 5, we see similar performances of SVM as compared to MLP, though with much shorter training times. SVM time complexity is predominantly bounded by the number of pairwise distances between samples over each feature but that may be outweighed by the computational intensity that arises from the forward pass and backpropagation of an MLP.
- Aside from that, the degree of non-linearity can be inferred – we see non-linear kernels generally outperforming linear ones and a polynomial kernel with order 5 performing the best. Non-linear kernels and those with higher orders allow for more of the non-linear relationship to be captured though, at the cost of longer training times and potential risk of overfitting.
- Larger regularisation parameter values indicate our distaste for misclassified samples (i.e. imposes a larger penalty) but at the cost of a smaller margin. But since our validation set was sampled from the overall training set, the smaller margin may not be detrimental. Indeed, we generally see an SVM's performance to improve with larger regularisation parameter values.
- The best performing SVM had an f-score of 70.08% and training time of 116.11 seconds.

CNN

- From Figure 6, we see that our CNN's performance varied marginally over different hyperparameter combinations, in terms of both f-score and training time. The kernel acts as a filter to extract features from images but varying the kernel size didn't seem to result in drastic differences in performance.
- That said, the training times are significantly longer than both MLP and SVM – around 5 and 10 times longer respectively. On top of the computational complexity similar to an MLP, the convolutional layers and feature learning adds additional time complexities.
- Nonetheless, the best performing CNN had an f-score of 65.89% and training time of 1300.13 seconds but performed the worst when compared to our best performing MLP and SVM models.

Model Testing and Selection

- From Figure 7, we see similarities of model performance over our testing set. Though, SVM performed marginally better with more correct predictions over most classes, as expected.
- Additionally, we see predictions from our SVM to centre around the majority classes more, as compared to MLP and CNN. For example, though there were 74 images labelled as "fear", SVM only predicted "fear" labels 28 times as compared to 54 and 62 times with MLP and CNN respectively. Contrasting this to the "neutral" label which had 680 images assigned to it, SVM predicted images to be "neutral" 717 times as compared to 651 and 697 times with MLP and CNN respectively. In essence, our SVM forgoes false positives in the minority classes for false positives in the majority classes.
- One may view it as picking the 'low-hanging fruit' in the form of focusing on majority class prediction but this could be due to the distribution of the dataset itself. In Figure 2 we've seen how the training set is imbalanced and with it comes biases in predictions. For example, consider the extreme case where 99% of images are tagged as "neutral" and the remaining 1% is distributed over the remaining 6 classes, then a model which solely assigns "neutral" labels will be awarded an accuracy of 99%.
- In Figure 8, we visualise the trade-off between models. For example, in the first image both MLP and SVM correctly classifies the image as "sadness" but in the second image, only CNN classifies the image correctly with both MLP and SVM wrongly classifying the image as "surprise".
- Nonetheless, SVM is selected as our final model due to its performance across the 3 measures over the validation set, better performance over the testing set and significantly shorter training times.
- In Figure 9, we again we see a mixture correct and incorrect predictions from our model. The SVM correctly predicts the first and third frame as "happiness" and "sadness" respectively but misclassifies a "neutral" face as one that shows "sadness" instead.