Deep Learning (COSC2779)

Assignment – 1

Student Name: Salina Bharthu Student No: S3736867

Contents

Table of Figures:

Objective

The objective of this document is to outline the approach and results of deep convolutional neural network (CNN) developed to estimate the head pose of a person's image.

Introduction

As a part of this assignment, we are given a dataset containing 2790 head pose images of 15 persons. The task is to predict two angels of the head pose, that are tilt (vertical angle of head) and pan (horizontal angle of head), using deep CNN.

To implement this, two separate classification models are developed to predict the tilt and pan angles of an image. Image data is explored and pre-processed to feed into neural network. The base line model with few CNN layers and fully connected layers is implemented for initial checks. After that, the baseline CNN model architecture is further modified as per the diagnostics of baseline model. The parameters of CNN architectures are tuned to further improvise the model performance. The model is trained using the obtained parameter set and evaluated on unseen data using metrics such as accuracy, precision, and recall that are easily interpretable for the classification model performance and further the model is used for prediction on unseen test data.

Methodology

Data Preparation

Initial Image Data Exploration: Initially, few images from the directory path are loaded and explored. The given .jpg images are converted into grayscale as the colour is not the differentiating factor in this task. Also, the suitable image size (100*100) that is interpretable and suitable for all images, is identified after experimenting with several image sizes.

Train- Validation-Test split: Two separate csv files containing the information of image filename along with person_id and series_id, are extracted as data frame. The training set contains 2325 images of 13 persons and test set is having 465 images of 3 persons. The test set is without label information and need to be used at the end to provide the predicted data.

Now, the training set have target labels (tilt and pan angles) and to understand the distribution of data among different labels of pan and tilt, the class frequency graph is plotted and inferred that, all the classes are having equal distribution except 90 and -90 tilt labels. The minor imbalance is further addressed by performing data augmentation.

First, the training set is divided into train and test(holdout) sets using person_id to ensure the ultimate model evaluation on unseen data (unseen person's image). The separated training set is further divided into train-validation (80%-20%) datasets. Here, the train and validation dataframes are splitted using stratify with target variable to ensure equal distribution of images from all classes.

Image Data Pre-processing: For Image data pre-processing, functions from Keras ImageDataGenerator class are used. The training and validation batches are prepared from the respective data frames. While performing this, the images are initially rescaled and converted into float representation between 0 to 1. This can increase the model efficiency in terms of runtime and resource utilization. Moreover, Images are shuffled to avoid possible class biases and further processed using initial batch-size of 16. To determine the suitable batch-size that offer less training time and better generalization, the tuning is performed at later stage.

After applying these basis pre-processing actions upon training and validation images, the separate batch of Augmented Images are created. The images are randomly resized, brightness is randomly changed and resized back to 100*100. This image data augmentation can be useful in-order to identify the important features of the images.

CNN Model Implementation

Baseline CNN Model: As CNN is widely known for its capability to craft important features from image data, in this task, first, the baseline CNN model is developed. The images with 100*100*1 size are given as an input. The baseline model with 2 (convolution 2D layer + max pooling) with filter size of 16 and 32 respectively, kernel size of (3,3) and 'relu' activation in convolutional layers, is created. The non-linear 'relu' activation is used as the image array have pixel size data between 0 to 1 (non-negative) and it can help accelerating the convergence in this scenario. Both max pool layers have pool size of (2,2). These layers are further followed by fully connected dense layer with 64 number of neurons and 'relu' activation. The output layer contains 9 (number of unique tilt angles) nodes for tilt model and 13 (number of unique pan angles) nodes for pan model along with 'softmax' activation that is suitable for multi class classification problems.
Moreover, the 'adam' optimizer is used as it provides adaptive learning rate and model is initially trained on 50 epochs. To evaluate the model performance, accuracy (idle for classification problems) and categorical cross entropy (loss function suitable for multi class classification task) metrics are used, and training and validation curves are plotted. The baseline model is trained over non augmented and augmented training batches.

After analysing the training and validation curves for both tilt and pan models (Figure 5 and Figure 6), it can be inferred that the model gives better accuracy and bias-variance trade-off while using augmented train data. The lower loss on augmented batches indicate the efficient extraction of features from augmented images. Therefore, augmented images are utilized for further implementation.

CNN Model Architecture Modification for Tilt Classification:

As inferred from the learning curves of baseline tilt classification model, the model gives reasonable results over augmented training batches with low generalization gap. However, to generalize well on the unseen data further, one more convolutional and max pooling layer is added. Also, the fully connected dense layers are added empirically based upon the computational units passes to the model. There is a minor overfitting and increasing the model complexity by adding CNN and dense layers can lead to overfitting, therefore, dropout layer and L2 kernel regularizer is added.

CNN Model Architecture Modification for Pan Classification:

As inferred from the learning curves of baseline pan classification model, the model is overfitting data and the generalization gap is high. Therefore, adding kernel regularizer in Convolutional layers and dropout layers in fully connected architecture. Also, to enhance the model performance in terms of accuracy, the additional 2 (convolution 2D layer + max pool layer) are added.

For faster and better convergence, the 'adam' optimizer is configured with decaying learning rate (Using exponential decay in learning rate helps in faster convergence) for both tilt and pan classification models. The initial configurations of the model architecture (such as filters, kernel size, pool size, dropout rate and learning rate) are considered after few trials manually and further tuned at later stages for both models. Initially the models are trained for 100 and 150 epochs and evaluated using training and validation curves of loss (categorical cross entropy for multiclass classification problem) and accuracy by epochs. The tilt classification model seems to converge well around 150 epochs and pan classification model around 100 epochs.

Hyper Parameter Tuning:

To further explore the best suitable parameter values as per the task for CNN model, the hyper parameter tuning is performed using randomizedsearchcv with 3-fold cross validation. For each cross-validation fold in randomizedCV, the random subset of all passed parameters are picked to use for model training and results are returned. (The in-depth explanation of the hyper parameter tuned using randomizedsearchcv and its significance is mentioned in .ipynb file).

Model Fitting, Evaluation and Judgement:

The best parameter set returned after hyper parameter tuning is utilized for further model training. The plot (Figure 1) for pan model shows that the minor signs of overfitting and fluctuations in validation curve. This result can be enhanced by adding more training data and experimenting more with regularization parameters. The plot (Figure 2) for tilt model shows the better convergence and stabilization, indicating the well model fit.

The holdout test data is utilized for final evaluation of the model as it contains images of unseen persons. Using classification report and confusion matrix for evaluation enables to understand model performance class-wise.

The pan model shows that approx. 57% of the instances are correctly identified across all 13 classes. The higher colour intensity on the diagonal of confusion matrix (Figure 3) shows that the model does not have high bias towards any specific class. The tilt model provides average accuracy of 59% in classifying the correct instances across 9 classes. The confusion matrix (Figure 4) shows that, tilt model does not classify the images with one particular label (0 tilt angle - 0 precision) correctly.

Conclusion

For this task, the evaluation can be done based upon the true positives, that is precision. Therefore, considering it for the ultimate judgement. The model predicts the head pose angles with average 60% precision. The accuracy of prediction seems to be low for Deep CNN model developed for this task, however, the models provide well balanced fit in-terms of generalization gap and bias-variance trade off.

Limitation and Future Work

The two separate classification models developed for this task can only predict the head pose angles that are present in the training data. For instance, if the image has tilt or pan angle of 5, then it will predict the nearest similar label from the available set of labels.

The model provide accuracy of roughly 60% on unseen data, which can be improvised by implementing transfer learning and doing more research in the application area.

References

1. *RMIT Portal*. Rmit.instructure.com. (2020). Retrieved 7 September 2020, from https://rmit.instructure.com/courses/67346/pages/week-5-learning-materials-slash-activities?module_item_id=2521536.
2. *tf.keras.preprocessing.image.ImageDataGenerator*. TensorFlow. (2020). Retrieved 7 September 2020, from

https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator#flow.

3. *Deep Learning*. Deeplearningbook.org. (2020). Retrieved 7 September 2020, from https://www.deeplearningbook.org/.

Appendix

1. Final Model Training and Validation Curve

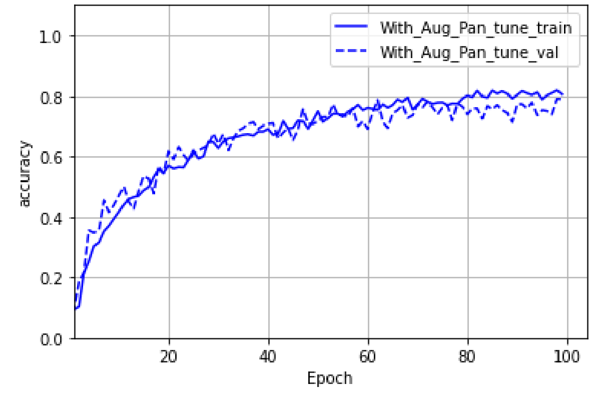


Figure 1 - Pan Classification training and validation curve

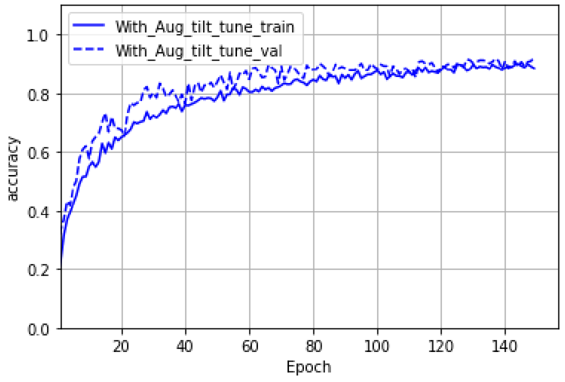
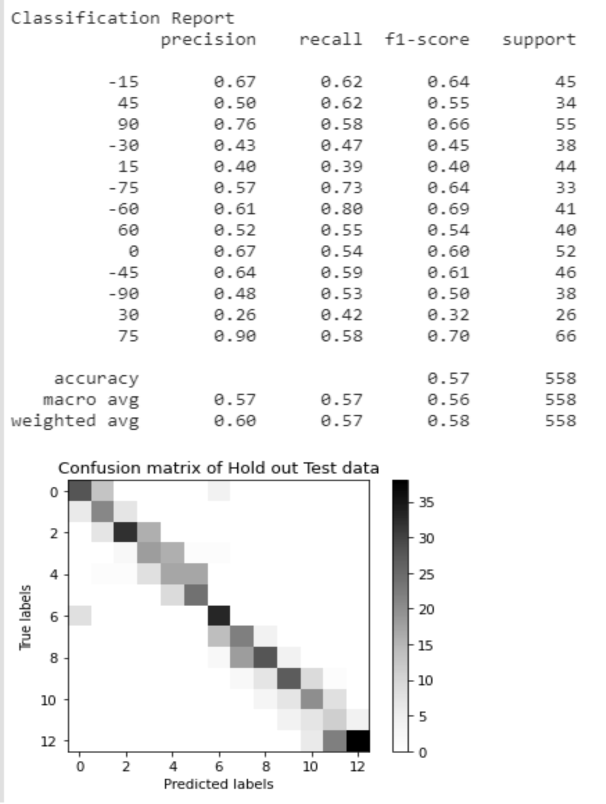


Figure 2 - Tilt Classification training and validation curve

2. Classification report and Confusion Matrix

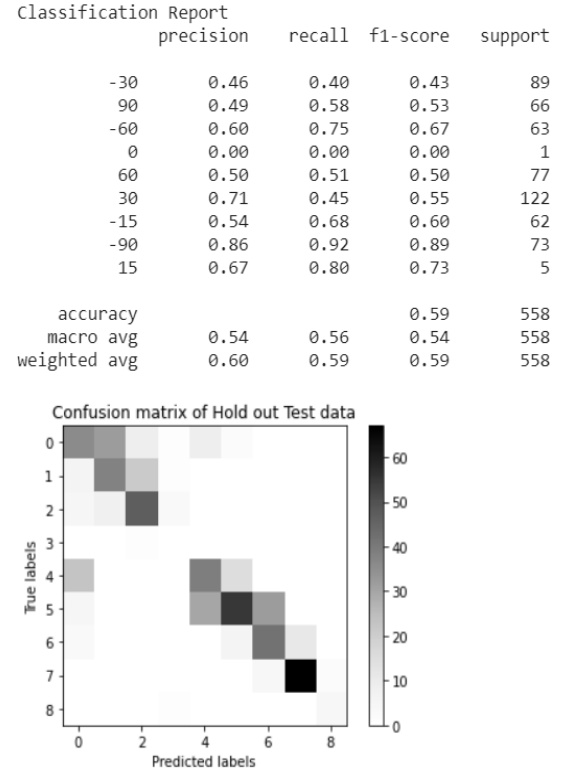


```
Classification Report
              precision    recall  f1-score   support

         -15       0.67      0.62      0.64        45
          45       0.50      0.62      0.55        34
          90       0.76      0.58      0.66        55
         -30       0.43      0.47      0.45        38
          15       0.40      0.39      0.40        44
         -75       0.57      0.73      0.64        33
         -60       0.61      0.80      0.69        41
          60       0.52      0.55      0.54        40
           0       0.67      0.54      0.60        52
         -45       0.64      0.59      0.61        46
         -90       0.48      0.53      0.50        38
          30       0.26      0.42      0.32        26
          75       0.90      0.58      0.70        66

    accuracy                           0.57       558
   macro avg       0.57      0.57      0.56       558
weighted avg       0.60      0.57      0.58       558
```

Figure 3 - Classification report and confusion matrix for Pan Model



```
Classification Report
              precision    recall  f1-score   support

         -30       0.46      0.40      0.43        89
          90       0.49      0.58      0.53        66
         -60       0.60      0.75      0.67        63
           0       0.00      0.00      0.00         1
          60       0.50      0.51      0.50        77
          30       0.71      0.45      0.55       122
         -15       0.54      0.68      0.60        62
         -90       0.86      0.92      0.89        73
          15       0.67      0.80      0.73         5

    accuracy                           0.59       558
   macro avg       0.54      0.56      0.54       558
weighted avg       0.60      0.59      0.59       558
```

Figure 4 - Classification report and confusion matrix for Tilt Model

3. Training and Validation curves of BaseLine CNN models

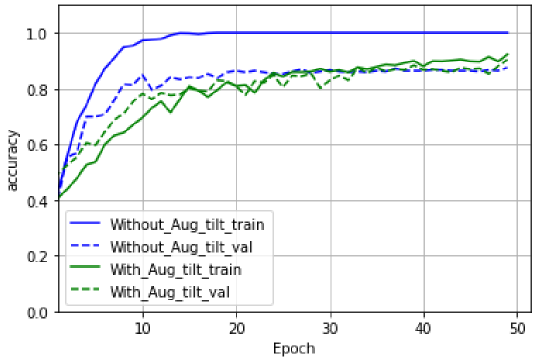


Figure 5 - Baseline Model Tilt

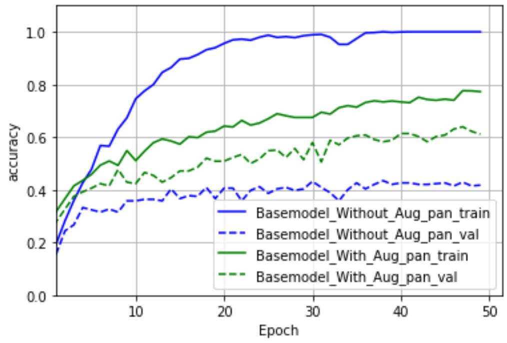


Figure 6 - Baseline Model Pan