Head Pose Estimation

1. Abstract

The main objective of this report is showcase the study behind discovering the best fitting machine learning model which provides best head pose estimation on two factors, tilt and pan. Approaches using CNN proved to be most effective in this estimation. For testing and training purpose we use the Head Pose Image Database , the performance of the model is derived keeping recall and accuracy as the performance evaluator, primary evaluator is the plots, which helped in determining cases of under fitting and over fitting .Analysing of intra variability of classes and comparing performance of Convolutional Neural Networks based on parameter tuning provided the optimal model. The images were grey scaled as colour was not a determining factor tilt and pan estimation. Images were sized to a 32*32 density. Our chosen model for tilt consist of 3 CNN layer of kernel size 64,64 and 128 and one fully connected layer of 128. Over fitting was the major issue we faced here and a dropout under each layer help mitigate this issue.The model was chosen on basis of good training to validation curve and accuracy score of 85 with recall 69. Our chosen model for pan consist of quite more number of layers mainly due to 13 classes present in pan attribute. It contains 4 CNN layers of kernel size 32,32,64 and 64. A dense value of 128 is chosen for final MLP layer ending with softmax to provide more probable output feature value

2. Introduction

Head poses is an important medium of non verbal human communication thus plays an important role in human-computer interactions. Proliferation in Human computer interface has made , head pose estimation a booming field mainly because of its wide spread application in human behaviour analysis, gaze estimation systems or driver assistance systems as head pose provides direct information of a persons attention [1]. Head pose estimation has significant impact in the field of gaze estimation too, which has facilitated human interaction with machines. The standard computer vision techniques employed could not succeed mainly due to the demand of huge manual work in form of feature extraction [2]. Deep learning was used to alleviate this drawback. The concept behind deep learning, also known as artificial neural networks exist forasmuch as the early 1940 [3] but its only after AlexNet wins ImageNet in 2012 that it became an extensive research field with wide applications.Before the arrival of neural network in field of human pose estimation work was mainly done via pictorial structures [4] and sophisticated extensions [5] which lacked accuracy. fully connected CNN architectures based on heatmap regression[6] has reformed human pose estimation giving high accuracy for even most challenging dataset.

3. Dataset

The data for this assignment is the modified version of Head Pose Image Database. The database provided has 2790 monocular face images of 15 persons with different pan and tilt angles [7] . Quick analysis by random generation of images provided info about factors of people in data base. Varying factors include People in the database wear glasses or not and has various skin colour. Background is neutral to avoid noise.The data is given in the form of colour images hence giving 3 layer composition, which are then resized 32x32 uniformly. The images are in JPEG format and the zip file weight a 11.3 mb on disk. On preliminary analysis of data few concerns were raised, which are as follow.

Concern 1 : The spread of the data with regards to tilt was done to find a very less weightage to -90 and 90 tilt angle data, as shown in Fig 1.1 and Fig 1.2. Applying same weighing technique over pan we saw an uneven weightage trend there too with maximum number of images available for 0 pan as shown in Fig 2.1 and Fig 2.2. This variability could lead to improper training of model which could give biased outcomes towards image category that are higher in number.

Concern 2 : Random selection of images from the pool showed distorted images in the form of incomplete face details in the image as shown in Fig 3. The price for this abnormality when applying Image analysis techniques is improper training of model giving false prediction. Also the

span of face in image seems considerably different, which gives different structural size difference to person face features.

4. Methodology

The primary parts of our algorithm is Dataset, Cost function, Model and Optimization procedure.The tenses we operate on here is the width the height and the channels of the images, which is 32*32*3. Since we are grey scaling the images, theoretically we have only one channel.

Our goal is to design an algorithm which will improve the weights $\mathbf{w}^*$ = argmin L ($\mathbf{w}$) in such a way that reduces the MSE of test set when model is let to train on training set, mathematically, find the max value of weights θ for data D in hypothesis $h^*(\mathbf{x})$. $p_{model}(\mathbf{x}; \theta)$. Theoretically we do so my feeding weight value into model, calculate associated loss using data and take min value and use $\mathbf{w}^*$to represent optimal hypothesis.The Task or unknown target function: y = f (x) here is to predict the head pose in terms of tilt and pan value of a given person from input image by mapping pixel value to head pose. Feature of the task: $\mathbf{x} \in R^d$ Is the pixels in the images given. We are using sequential api to built our models.Our model consists of hidden layers(the layers between input and output layer), .this consists of two parts, $\mathbf{z}^{(i)} = \mathbf{W}^{(l)}\mathbf{x}^{(i)} + \mathbf{b}^{(l)}$ which is affine transform and $h^{(i)} = g\ \mathbf{z}^{(i)}$ activation part we try with sigmoid activation function in but its not ideal due to saturation issue. We need to administer non linearity while doing the convolution and stacking the weights since convolution operation in itself is a linear function , therefore to make decision boundary's non linear we have to apply non linearity activation function. These activation functions are used to convey the errors effectively, popular activation function include Sigma, Than and Relu .

5. Modelling

We split the data into training, test and validation set using train_test_split( ) which splits the arrays or matrices into random train and test subsets, this facilitate checking of generalisation error.

TILT :

Taking into consideration Universal Approximation Theorem (Horniket al., 1989; Cybenko, 1989), we are initially taking just one layer to built a CNN model. We have 32x32 features and nine classes, so the input layer must have 1024 units, and the output layer must be of 9 units. We define the hidden layers . We're only going to have one hidden layer for this model, with 64 neurons. clearly this gave us a clear case of overfitting , seen from the plot where the training super comprehends the validation proving the test error is large while train error is small. Also we can comprehend that the dataset is large and more complicated to process with one layer, hence we are going to load data in batches into the memory, we will be using image data generators for this. Choosing batch size - larger batch will provide more accurate estimate of gradient but with diminishing return.amount of memory also increases. We should not use mini matches multiple times, the batch size chosen here is 16. In the loading function The training process could delay significantly, if we keep the default pixel value thus we need to bring it to range of 0-1we rescale this value by dividing it with 255, which intern is the maximum pixel range.

Now to increase the accuracy we try to increase the depth of the network. Theoretically increase in layer is supposed to give better accuracy as shown in Fig 6 but we need to see if this accuracy is due to increased capacity or actual increase in depth. Model 2 we try a three layered model with MLP as the final layer to help convergence. we are using categorical_crossentropy since the output variable contains nine classes . we try with sigmoid activation function but its not ideal due to saturation issue, thus reLu Fig 4 is being used, although this too has the issue that the units can die when input is towards negative side and learning becomes zero, thus we initialise bias to small positive value.Since we have multiple output classes we are going to use softmax where we find the probability of each class as shown in Fig

5. We use the loss function $\mathcal{L}(\mathbf{w}) = -\sum_{i=1}^{N}\sum_{j=1}^{c} y_j^{(i)} \log\left(\hat{y}_j^{(i)}\right)$ which will encourage all other less probable values to be stepped down.
The cost function associated here is non-convex,looking at the plots we can see that If it is too large, the learning curve will show violent oscillations, and often increasing significantly. If the learning rate is too low, learning proceeds slowly as in SGD used . Usually we have a high learning rate at onset and reduces as per decay policy, a large learning rate at start allow to explore a big area in loss function and then towards end it converges to a particular value. We use Adam over SGD because of its ability to adjust learning rate during optimisation process. It does it by computing moving average of square gradient then calculate moving average of gradient then apply the two components to update the weight. Finally the training and validation accuracy of without dropout is not in sync. Thus applying dropout value 0.5 reduces the complexity of model and a 3 CNN and 1 FCN with Adam optimiser gave a recall value of 76 and accuracy of 82. Fig 7.1, 7.2 and 7.3 shows the plot, confusion matrix and accuracy data of chosen model(model 8).

PAN

Pan models have 13 classes to action, because of such high class numbers we used higher number of layers. We used regularisation which is used to create a ridge penalty, The main idea is to decrease the parameters value which in turn should reduce the variance .-Pooling - This will reduce the feature map size , each time a pooling happens the feature map becomes smaller and smaller until a flattening happens categorisation is reached . This change in pooling value did not yield a desirable result, in fact had a validation line hovering over the test line,
Non - linearity - relu function is the most default function used, change of relu to sigmoid function yield an error rate because in sigmoid function for two different inputs a single output is possible, thus it dent help in in-bedding non linearity. Since we have high number of classes we kept the batch size as 32 for pan analysis.The initial model tested was a simple sequential model with l2 regularizers of 0.001 , this produced an output plot which portrayed proper learning curve initially but then training set followed a linear path profile while validation did not.The following parameters were fine tuned

- Depth: Depth is determined by number of filters, here in model_9 there are 5 depths while model 5 has 1 .From recall value we can clearly see model_9 outperforms model 5.
- Stride: It is the distance to move the input over the centre of each filer, default value is 1 is chosen since class value does not seem to be changing.
- Activation Function : the most commonly used activation function is “relu” as its proven to mitigate the saturation problem we saw in model 5.
- Convolutional kernel value - we can choose kernel values as 32-32-64-64-128-128, which provided us with maximum accuracy and recall value.

Fig 8.1,8.2 and 8.3 shows the plot, confusion matrix and accuracy data of chosen model.

---

6. Ultimate Judgement

Use of feed forward neural networks like CNN provides finer solution since convolutional layers reduces the number of weights significantly also practices like rectified linear units that pass on errors more efficiently.The uneven spread of data weight both in tilt and pan had its implications with more effect seen in tilt due to less weight over -90 and 90 angles. The resolution of images provided had spread issues and thus hiding important information required to decipher pan and tilt attributes. The limitation due to datasets size and image resolution could be solved with larger training datasets of images with a higher resolution. Use of pre trained models like ResNets [8] could provide much better accuracy but also requires high computational power and the current image resolution could have not been adequate . Here a high range value could lead to head pose to be of minimum attention span which could be dangerous if scenario is of driving . Throughout our model building we saw a large False Negative value output thus choosing recall as the performance measure criteria. For Tilt model, we had major issue with model getting over fit, we tried conventional image augmentation but it threw away bad result mainly due to facial feature alignment getting destroyed. Use of dropout with a 0.5 wait time made the neurons not learn values from input in training stage as a percentage of each neurons in hidden layer is randomly deactivated. Also we used Adam as optimiser and relu as activation function for each layer and softmax for final layer. For Pan we choose a deep layered model since we have large class attributes, we did MaxPooling that select the maximum value from the 2 dimensional region, thus reducing the dimension. We have two dense layers of 128 configured in fully connected final layers, with flattening that converts our 3D feature maps to 1D feature vectors.

8. Appendices

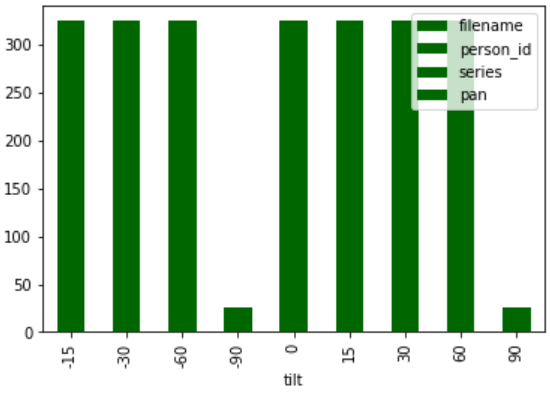

Fig 1.1

tilt	filename	person_id	series	pan
-15	325	325	325	325
-30	325	325	325	325
-60	325	325	325	325
-90	25	25	25	25
0	325	325	325	325
15	325	325	325	325
30	325	325	325	325
60	325	325	325	325
90	25	25	25	25

Fig 1.2

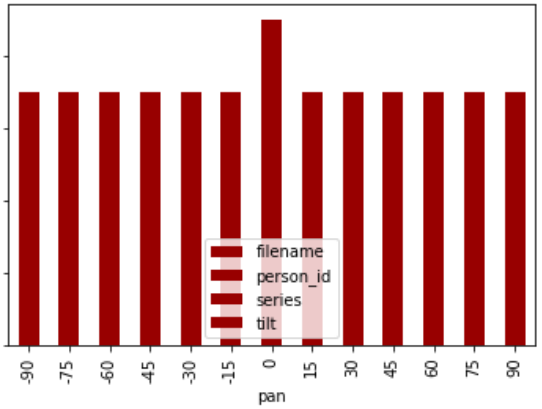

Fig 2.1

Fig 3

pan	filename	person_id	series	tilt
-90	175	175	175	175
-75	175	175	175	175
-60	175	175	175	175
-45	175	175	175	175
-30	175	175	175	175
-15	175	175	175	175
0	225	225	225	225
15	175	175	175	175
30	175	175	175	175
45	175	175	175	175
60	175	175	175	175
75	175	175	175	175
90	175	175	175	175

Fig 2.2

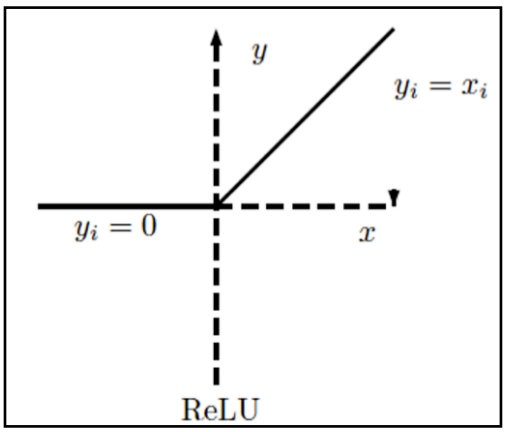

Fig 4

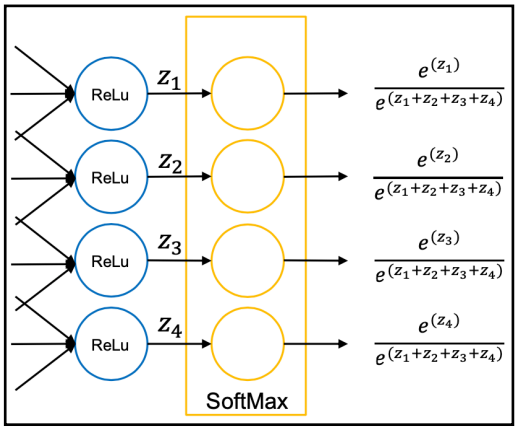

Fig 5

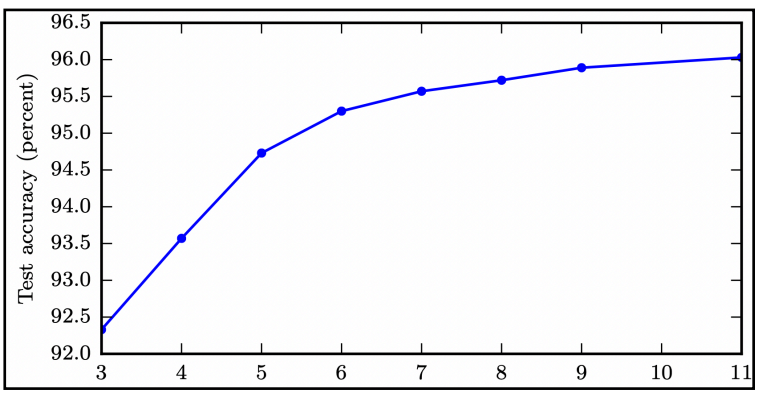

Fig 6

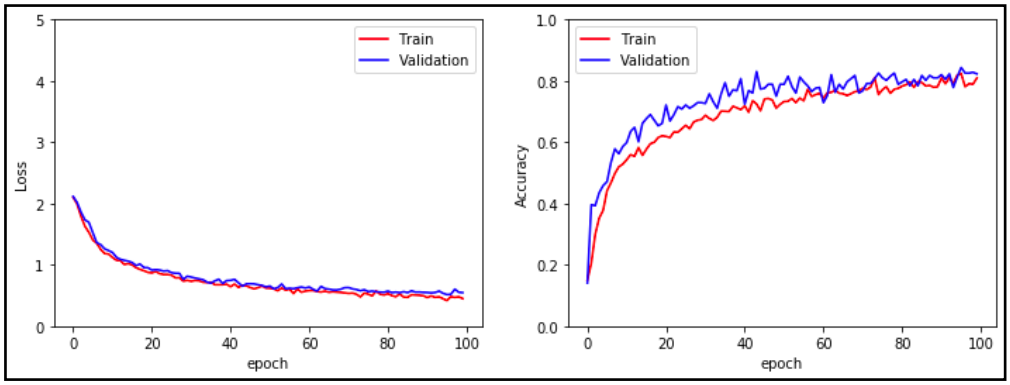

Fig 7.1

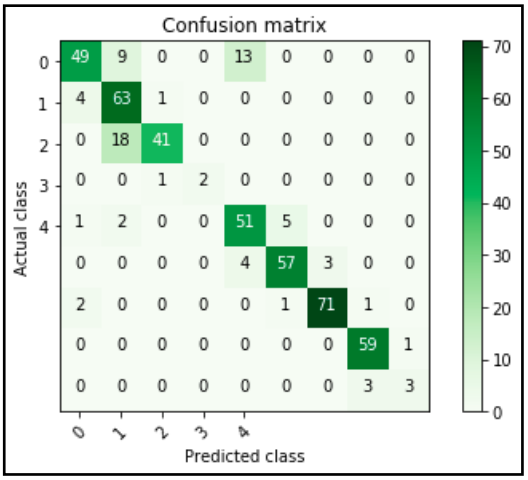

Fig 7.2

Prediction shape is (465, 9)

	precision	recall	f1-score	support
-15	0.88	0.69	0.77	71
-30	0.68	0.93	0.79	68
-60	0.95	0.69	0.80	59
-90	1.00	0.67	0.80	3
0	0.75	0.86	0.80	59
15	0.90	0.89	0.90	64
30	0.96	0.95	0.95	75
60	0.94	0.98	0.96	60
90	0.75	0.50	0.60	6
accuracy			0.85	465
macro avg	0.87	0.80	0.82	465
weighted avg	0.87	0.85	0.85	465

Fig 7.3

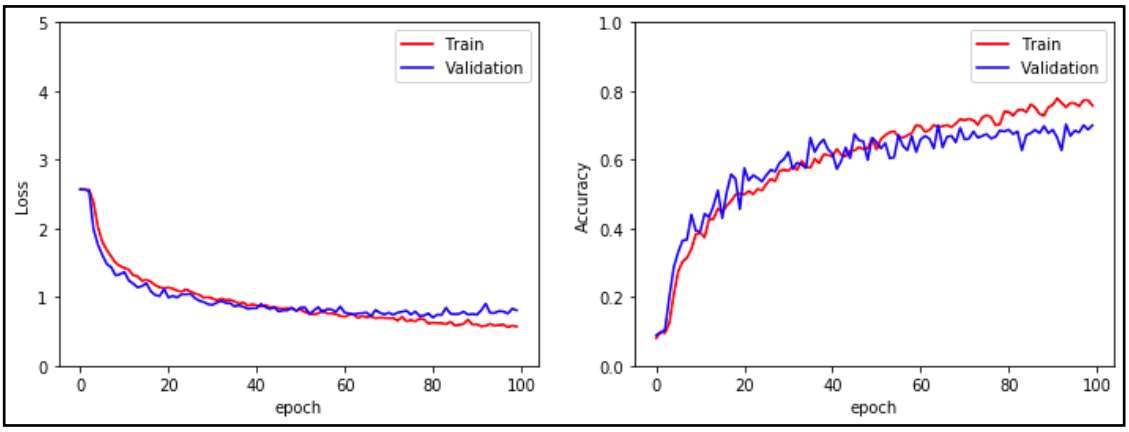

Fig 8.1

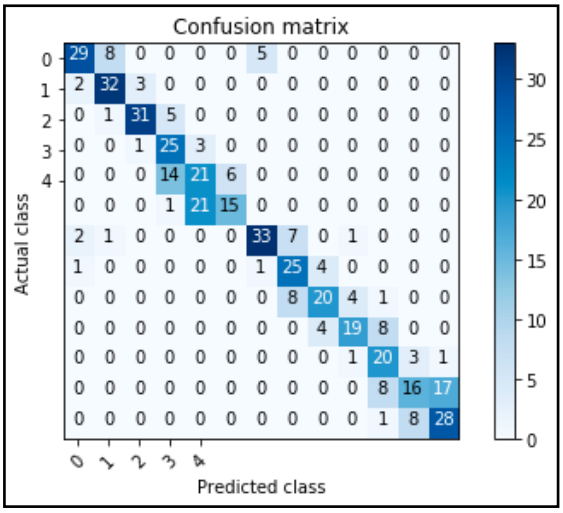

Fig 8.2

Prediction shape is (465, 13)

	precision	recall	f1-score	support
-15	0.85	0.69	0.76	42
-30	0.76	0.86	0.81	37
-45	0.89	0.84	0.86	37
-60	0.56	0.86	0.68	29
-75	0.47	0.51	0.49	41
-90	0.71	0.41	0.52	37
0	0.85	0.75	0.80	44
15	0.62	0.81	0.70	31
30	0.71	0.61	0.66	33
45	0.76	0.61	0.68	31
60	0.53	0.80	0.63	25
75	0.59	0.39	0.47	41
90	0.61	0.76	0.67	37
accuracy			0.68	465
macro avg	0.69	0.68	0.67	465
weighted avg	0.69	0.68	0.67	465

Fig 8.3

9. References

[1] A. a. R. MikelAriz, "Robust and accurate 2D-tracking-based 3D positioning method: Application to head pose estimation," Computer Vision and Image Understanding, vol. 180, pp. 13-22, March 2019.

[2] M. A. A. M. A. H. A. H. S. A. S. P. J. K. a. M. B. M. Safat B. Wali, "Vision-Based Traffic Sign Detection and Recognition Systems: Current Trends and Challenges," NCBI, 6 May 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6539654/. [Accessed 2 September 2020].

[3] D. Kriesel, "A Brief Introduction to Neural Networks," 2007. [Online]. Available: http://www.dkriesel.com/en/science/neural_networks. [Accessed 3 September 2020].

[4] Felzenszwalb, P.F., Huttenlocher, D.P. Pictorial Structures for Object Recognition. International Journal of Computer Vision 61, 55–79 (2005). https://doi.org/10.1023/B:VISI.0000042934.15159.49 [Accessed 4 September 2020].

[5] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," CVPR 2011, Providence, RI, 2011, pp. 1385-1392, doi: 10.1109/CVPR.2011.5995741 [Accessed 5 September 2020].

[6] A. B. a. G. Tzimiropoulos, "Human Pose Estimation via Convolutional Part Heatmap Regression," in European Conference on Computer Vision, 2016. [Online]. Available: https://www.researchgate.net/publication/307897694_Human_Pose_Estimation_via_Convolutional_Part_Heatmap_Regression [Accessed 5 September 2020].

[7] D. H. J. L. C. N. Gourier, "Head Pose Image Database," [Online]. Available: http://www-prima.inrialpes.fr/perso/Gourier/Faces/HPDatabase.html. [Accessed 6 September 2020].

[8] I. S. Rieger, "Head Pose Estimation using Deep Learning," 2018.