1. Introduction

Pose estimation refers to the problem of localization of joints in images or videos. It can be divided into 2D pose estimation, where the task is limited to an x-y plane, and 3D pose estimation, where the task is expanded to an x-y-z plane. Pose estimation is a difficult problem in practice because of lighting changes, obstructions, and barely visible joints. However, it is an important problem because it offers various applications in animation, gaming, and action recognition.

Deep Learning approaches to solving this problem began with "Deep Pose" by Toshev et al, wherein the problem was formulated as a CNN-based regression problem. The model used an AlexNet backend of 7 layers with a final layer consisting of 2x joint coordinates, where x is the number of joints. This paper was significant as it is said to have kickstarted research in this area. However, several improvements have been made since then, and the HRNet paper has been the most successful of all. Previous research had adopted neural network layers going from high to low to high image resolution. HRNet maintained the high resolution throughout the architecture which resulted in a strong performance.

This assignment will make use of MMPose, a popular open-source library used for a wide variety of pose detection tasks. MMPose is based on PyTorch and has been developed as a part of the OpenMM Lab project. The OpenMM Lab project prides itself as the "most influential open-source computer vision algorithm system in the deep learning era". This claim is statistically supported by their 250+ algorithms, 2000+ pre-trained models, and 1000+ contributors spanning over 20 different research fields. The project consists of different areas of computer vision such as object detection, image classification, and 2D and 3D pose estimation to name a few.

The goal of this assignment is to develop a top down 2D cat pose estimation system. The idea is to detect body keypoints of the cats (17 in this case to be precise) and make use of state-of-the-art deep learning techniques to predict the pose. The following sections will describe the methodology and provide a detailed walkthrough of how the assignment was carried out and implemented. Improvements will be made to the baseline model by hyper-parameter tuning, selecting different neural network architectures, and trying various data augmentation techniques. Experiments and analysis will be conducted and will include dataset analysis, model evaluations and comparisons through data augmentation.

2. Methods to Improve Model Performance

2.1 Neural Network Architectures and Transfer Learning: mmpose has a lot of backbone models available that have been pre-trained on different datasets such as Coco val2017, AnimalPose, and AP-10K. These models are based on the ResNet or HRNet architectures and some of them consist of pre-trained models. Using these models as a starting point allows for much faster training- the models can achieve a mAP of over 0.4 on the validation set after training for just 10 epochs. Below is a summary of the models experimented with using the parameters provided in their respective config files. Each model is trained for 20 epochs, and the best performing model- hrnet_w32_coco_256x192, is selected for further tuning. All the models tested here will be later used when performing experiments.

Model	PreTrained	mAP
Res50_ap10k_256x256	torchvision://resnet50	0.429
resnetv1d152_coco_256x192	resnet152_v1d_coco_256x192	0.437

Res101_ap10k_256x256	torchvision://resnet101	0.447
hrnet_w48_ap10k_256x256	hrnet_w48_ap10k_256x256	0.537
hrnet_w32_coco_256x192	hrnet_w32_coco_256x192	0.648

2.2 Data Augmentation Techniques

Data Augmentation has proven to be useful in reducing overfitting by creating more data by taking the training data through various augmentation procedures. The ones used have been listed below along with the tuned hyperparameters:

i. Random Flip: Flips the train images. Flip probability= 0.5

ii. Half Body Transformation

iii. Random Scale Rotation: Rotates the training images. Rotation factor = 60, Scale factor = 0.5

2.3 Hyper-Parameter Tuning:

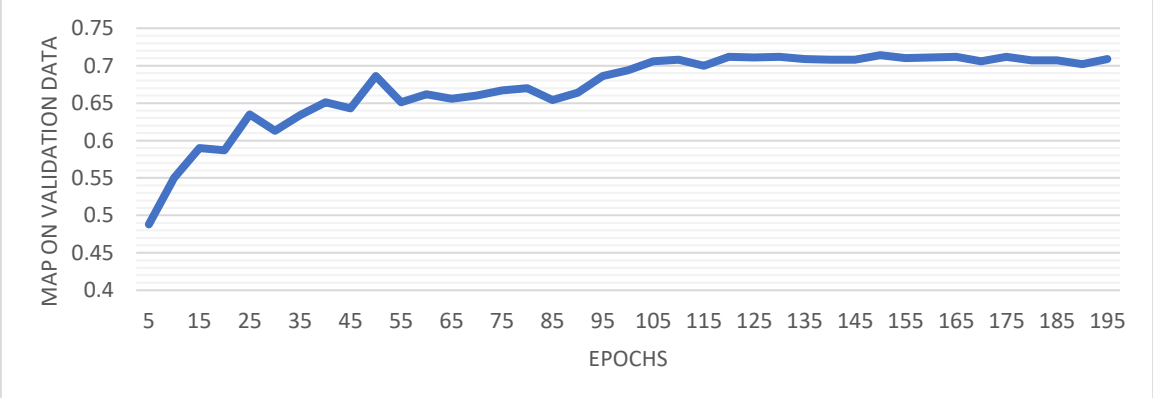
i. Learning Rate: Different Learning Rate values are tested for 20 Epochs using Adam and based on the results provided in the table below, 5e-04 is selected.

Learning Rate	5e-02	5e-03	5e-04	5e-05
mAP	0.155	0.364	0.648	0.559

ii. Learning Policy: The default learning policy is selected as all the backbone models available on mmpose are using it. Different hyperparameters will be tested in the next section to see if this standard policy works the best. For now, a step policy with a linear warmup, 500 warmup iterations, and 0.001 warmup ratio is selected. However, a step of [90,120] is selected since the number of epochs selected will be smaller than the 210 epochs chosen by the backbone config. This means that we start with a learning rate of 5e-07, which is incremented equally every step to reach 5e-04 after 500 warmup steps. Further, as per the step, the learning rate is divided by 10 at epoch 90 and 120.

iii. Optimizer: Using the tuned values from above, Adam, SGD, and RMSProp are tested as optimizers for 20 epochs. Adam offers the most promising results (compared to a mAP of 0 and 0.616 for SGD and RMSProp respectively) and hence is chosen as the Optimizer for updating the weights.

iv. Training Epochs: As seen below, increasing the number of epochs initially results in significant improvements in the mAP. However, after 120 epochs, the mAP result is almost a constant value, fluctuating between 0.7 and 0.715. Thus, the number of epochs is set to 150, giving the model some more training opportunity after the 2nd step taken at 120 epochs as defined in the learning policy above.



3. Experiments and Analysis

3.1 Dataset

The dataset provided contains 1328 images- 1000 of these are used for training, 128 for validation and 200 for testing. This results in a train:val:test ratio of 125:16:25. The annotation data has also been provided which consists of ground truth annotations and follows the COCO data format. Each cat consists of 17 keypoints which help identify the pose of the cat.

3.2 Experimental Setting

The different neural network architectures tested in section 2.1 are used for carrying out experiments. The weights are updated by the Adam optimizer. The learning rate is set as 5e-04 and divided by 10 at 90 and 120th epoch respectively. Training process is terminated at 150 epochs. 2 different input sizes 256x256 and 256x192 are used in the experiments. The experiments study the effect of various data augmentation techniques on the model performance.

3.3 Ablation Studies

3.3.1 Impact of Increasing Probability of Random Flip on Model Performance

Backbone	**Input Size**	**mAP @ Probability=0.2**	**mAP @ Probability=0.4**	**mAP @ Probability=0.6**
ResNet-50	256x256	0.591	0.604	0.60
ResNet-152	256x192	0.619	0.614	0.604
HRNet-w48	256x256	0.644	0.647	0.646
HRNet-w32	256x192	0.708	0.733	0.712

3.3.2 Impact of Rotation Factor on Model Performance

Backbone	**Input Size**	**mAP @ Rotation =40**	**mAP @ Rotation = 60**	**mAP @ Rotation = 80**
ResNet-50	256x256	0.582	0.603	0.596
ResNet-152	256x192	0.622	0.626	0.622
HRNet-w48	256x256	0.633	0.658	0.656
HRNet-w32	256x192	0.707	0.733	0.711

As seen in the 2 tables above, increasing the Probability of flipping first increases the mAP on the validation set as you go from 0.2 to 0.4. But when you move to 0.6, the value ends up decreasing. This could be because the model doesn't get to learn enough data and is presented with too much variety as the flipping probability goes up.

3.3.4 Impact of Adding Half Body Transformation

Backbone	**Input Size**	**mAP with Half Body Transform**	**mAP without Half Body Transform**
ResNet-50	256x256	0.582	0.594
ResNet-152	256x192	0.622	0.608
HRNet-w48	256x256	0.633	0.629
HRNet-w32	256x192	0.707	0.712

The addition of Half Body Transform offers a mixed result- some models perform better with Half Body Transform while some perform better without it. In fact, it can be noticed that for the denser models, ResNet-152 and HRNet-w48, the performance decreases by including Half Body Transform However, for ResNet-50 and HRNet-32, the model performance is seen to improve. This seems to suggest that model

complexity seems to have some relation with this data augmentation technique, however more research and analysis will be needed to draw out strong conclusions.