Detecting Human Poses from Still Images

Tharushi Suwaris and **Grishma Vadadoria**
Department of Computer Science
University of Aberdeen
52426304, 52428297
{t06ts24, grishma.vadadoria}@abdn.ac.uk

Abstract

This project investigates pose classification (sitting, standing, walking/running) using deep learning models on two datasets: original RGB images and background-removed images. We evaluated baseline models (FNN, CNN), a data-augmented CNN, a transfer learning approach (MobileNetV2), and CLIP-based multimodal embeddings. Results show that transfer learning on background-removed images achieved the highest test accuracy (71.43%) and F1 score (0.71). Experiments highlight the impact of model selection and preprocessing on classification performance. We also propose a future direction using pose estimation models like ViTPose for improved feature extraction.

1 Introduction

This project addresses a three-class pose classification problem; sitting, standing, and walking/running, using RGB images. We experiment with multiple model types, including a baseline feedforward neural network (FNN), convolutional neural network (CNN), augmented CNN, a MobileNetV2-based transfer learning model, and multimodal embeddings from the CLIP model. The goal is to evaluate performance differences across models and datasets, specifically comparing original and background-removed images. We also explore the generalization capabilities of each model, assess statistical significance, and analyze per-class classification performance.

2 Description of Data and Methods

This section outlines the dataset used, the preprocessing steps undertaken, the modeling choices made, and the overall experimental design. The goal is to carry out a supervised learning process to categorize images based on poses, which often contain complex visual information. We detail the dataset's origin and characteristics, the rationale for preprocessing and cleaning, the machine learning models explored, and the evaluation strategy used to assess performance.

2.1 Data

The dataset used in this project is a subset of 285 images from the MS COCO dataset. The goal is to classify the images based on the physical state of the people in the images, which is sitting, standing and walking or running. The original dataset includes complex real-world scenes with a wide range of visual information; some images contain a single person while some had groups of people. Additionally, most images have substantial amount of background content that may distract from the subject of interest.

Image property Analysis

To better understand the characteristics of the dataset, we explored the image properties by plotting the distributions of image width, height, and resolution (DPI). The minimum, maximum, and average values for these properties are summarized in Table 1. Figure 1 shows the distribution of these dimensions, which helped us identify outliers in image dimensions that could affect training consistency. Therefore, we followed resizing images before inputting them to the models to fast and better model learning.

Measure	Height	Width	DPI
Min	240	300	2,2
Max	640	640	4800,4800
Average	563	501	132,132

Table 1: Summary of image dimensions including minimun, maximum and average values of height, width and resolution in dots per inch

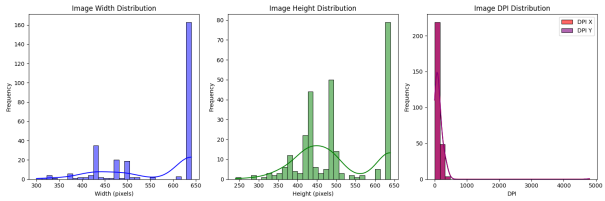

Figure 1: Histograms showing distribution of image heights, widths and DPI in the dataset.

Figure 2: 2(a)-Do not belong to any of the classes. 2(b,c,d)- Contains only a tiny part of body, making it difficult to classify to either of the classes.

Data Cleaning and Filtering

We also followed a data cleaning step to ensure the dataset was reliable and relevant. This includes checking incorrect labels, a manual review of the images was carried out to verify label accuracy. Images that were misleading or too ambiguous to confidently assign a class, such as those with only a tiny part of the person's body is visible or where pose was unclear was removed (Figure 2). Also, the images that did not clearly fit into any of the three classes were excluded to reduce the noise in the dataset. Following this,we have a dataset of 273 images and then we analyzed the class distribution to check for class imbalance, as imbalanced data can significantly affect model performance. A bar chart representing the number of samples per class is shown in (Figure 3). Based on the values, the class 'standing' has the minimum number for images (87 images) and class 'walking-running' has the maximum number of images which is 94. Considering the minority of the imbalance, we did not follow any oversampling or under sampling techniques.

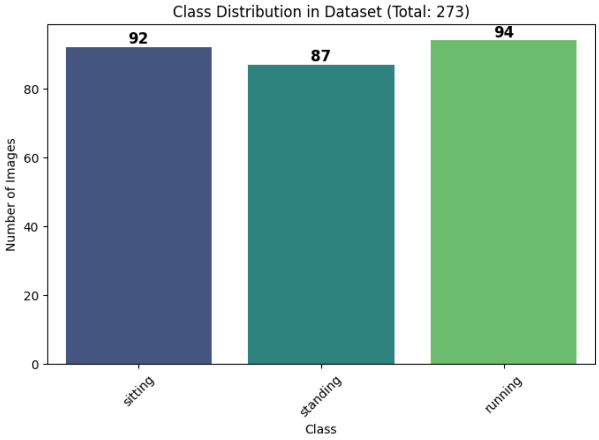

Figure 3: The number of images per classes

Background Removal

As another preprocessing step, we removed the background of the images saving only the person/people in the image. This was done using a pretrained deep learning model, DeeplabV3 with Resnet101[1] backbone, which is suitable for semantic segmentation tasks. The model was applied in evaluation mode to isolate human figures from the rest of the scene. This step was motivated by the high visual complexity of the original dataset, where rich backgrounds and contextual elements often diverted attention from the subject. By removing these distractions, we expect the model to focus solely on learning discriminative pose-related features. This not only helps simplify the input data but can also improve model generalization and reduce overfitting, especially when training on a relatively small data set. Some of the examples of before and after background removal are as in Figure 4. Having two datasets; original and background removed data, we can experiment how models behave with each input data.

2.2 Models/Algorithms

We explored a range of models for the pose classification task, including a baseline feedforward neural network (FNN), a convolutional neural network (CNN), and advanced models using regularization, transfer learning, and multimodal embeddings. Each model is described below with its design choices and training approach.

[1] https://pytorch.org/vision/main/models/generated/torchvision.models.segmentation.deeplabv3_resnet101.html

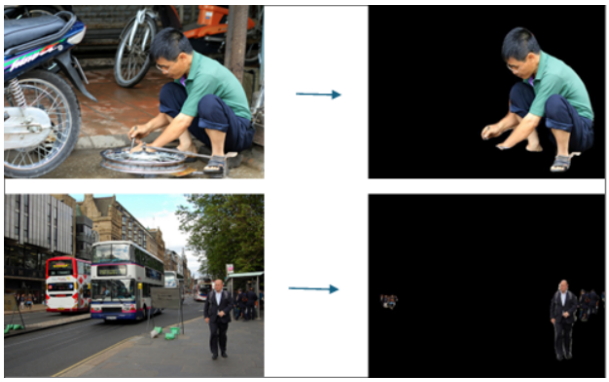

Figure 4: Exampels of before and after background removal

CNN Base

The baseline CNN accepted 3×224×224 RGB images and passed them through four convolutional blocks with channel sizes 3→32→64→128→256. Each block used 3×3 convolutions, batch normalization, ReLU activations, and 2×2 max pooling. The output feature map (256×14×14) was flattened and fed into fully connected layers, with dropout applied for regularization.

FNN Base

The baseline FNN took flattened RGB images (input size 150,528) and passed them through dense layers sized 150,528→512→256→128→3, with batch normalization, ReLU activations, and dropout. This simpler model served as a performance reference point for pose classification.

CNN-gen : CNN with Generalization Techniques (Data Augmentation)

To improve generalization, we applied data augmentation to the CNN model during training. Transformations included random rotations, brightness and contrast jitter, and affine translations using PyTorch's RandomRotation, ColorJitter, and RandomAffine. The architecture was identical to the baseline CNN but trained with augmented data to encourage robustness against overfitting.

Transfer Learning Model

We used MobileNetV2[2] [Mark Sandler, 2019], a lightweight model pretrained on ImageNet, known for its efficiency and strong feature extraction capabilities. Three strategies were explored: no freezing, partial freezing, and full feature extraction. The feature extraction approach, where all convolutional layers were frozen and only the classifier was fine-tuned, performed best and was selected for the final evaluation. This is a binary classification model, which is classifying classes sitting and standing.

Multimodal Embeddings 1: CLIP Classifier

The CLIP ViT-B/32[3] model was used to extract image embeddings. Each image was resized to 224×224 and passed through CLIP's visual encoder, producing normalized 512-dimensional embeddings. These embeddings were input to a lightweight FNN classifier consisting of an input layer, a hidden ReLU layer with dropout, and an output layer with three neurons for class prediction. This approach leveraged CLIP's pretrained visual semantics for efficient learning on limited data.

Multimodal Embeddings 2: Cosine Distance-Based Features

To enhance CLIP embeddings, we computed cosine similarities between each image embedding and textual embeddings for the three class labels ("a person sitting," "a person standing," "a person walking or running"). These three similarity scores were concatenated with the original 512 features, forming a 515-dimensional feature vector. A deeper neural network was trained on these enhanced features, using two hidden layers with ReLU and dropout. Feature importance analysis indicated that the cosine-based label similarities added meaningful discriminative power, improving classification performance.

2.3 Experimental Approach

The dataset was split into training, validation, and test sets to support robust model evaluation. For the CNN base, FNN base, CNN with data augmentation (CNN-gen), and the transfer learning model, a 70:15:15 ratio was used. Splitting was performed using torch.utils.data.random-split[4], ensuring approximate class balance through stratified sampling. In contrast, the models using multimodal embeddings (CLIP-based models) were trained with a 60:20:20 split, allowing more data for validation and testing given the lower data demands of pretrained embedding models.

[2] https://huggingface.co/docs/transformers/en/model_doc/mobilenet_v2

[3] https://huggingface.co/sentence-transformers/clip-ViT-B-32

[4] https://discuss.pytorch.org/t/torch-utils-data-dataset-random-split/32209

Two datasets were used in all experiments except multimodal embeddings: the original dataset of unaltered RGB images and a background-removed version, where DeepLabV3 with a ResNet101 backbone was used to segment the person and replace the background with black. All images were resized to 224×224 pixels, converted to tensors, and normalized using a mean and standard deviation of [0.5, 0.5, 0.5], following the standard normalization for pretrained models. These steps were implemented using torchvision.transforms.Compose to ensure consistency and model compatibility. Also early stopping criteria was used in all the models for better generalization. This setup allowed for direct comparison of model performance across different data preprocessing conditions, enabling investigation into how background removal and data augmentation affect classification accuracy.

3 Results

Test accuracy and F1 score were chosen as the primary evaluation metrics. Test accuracy provides a clear measure of overall model correctness on unseen data, while the F1 score offers a deeper understanding by balancing precision and recall, particularly in the presence of class imbalance. Together, they provide a comprehensive view of the model's performance. The final test performance of each model across both original and background-removed datasets is summarized in Table 2.

Model	Dataset	Accuracy %	F1 Score
CNN Base	Original	47.62	0.46
FNN Base	Original	35.71	0.36
CNN Base	Processed	38.10	0
FNN Base	Processed	26.19	0.26
CNN-gen	Original	38.10	0.38
CNN-gen	Processed	38.10	0.34
Transfer Learning	Original	54.76	0.44
Transfer Learning	Processed	71.43	0.71
CLIP Classifier	Original	65.0	0.64
CLIP + Cosine Similarity	Original	69	0.68

Table 2: Summary of test accuracies and F1 scores (macro). Note that 'Processed' dataset consist of background removed images.

A T-test[5] was performed to compare the CNN and FNN models, which confirmed a statistically significant ($p < 0.01$) difference only in originall data. This suggests that CNNs are more effective at leveraging spatial and contextual features present in the original images, which are largely absent in the background-removed version. When the background is removed, both models rely solely on foreground features, reducing performance and minimizing the difference between them. For complete T-Test, see Appendix A.

4 Discussion & Conclusion

The Transfer Learning model on background-removed images achieved the best overall results, with 71.43% accuracy and a macro F1-score of 0.71, suggesting that foreground-focused inputs paired with strong pretrained features can significantly improve classification. The evaluation metrices also suggested that this model performed well, somewhat balancing the classes.

CLIP-based models also performed strongly, particularly when cosine similarity with text prompts was added. The resulting classifier achieved 69% accuracy and an F1-score of 0.68.

Among the baseline models, the CNN on original images (47.62% accuracy, F1 0.46) outperformed the FNN (35.71%, F1 0.36). On background-removed data, both baseline models performed worse, with CNN scoring 38.10% accuracy and F1 0.00, and FNN 26.19% and F1 0.26, possibly due to reduced context from background removal. This implies that rather than the human pose models are affected from the other background information, which is possibly an incorrect approach. Even though it was expected to increase performance with background removed dataset, the reported performance suggests that the classical and simple neural networks are not that capable of differentiating human poses which often require to analyze human body parts. Also, these models seemed to be biased towards some classes having low F1 score and test accuracies, specifically, the CNN-gen with background removed dataset, which did not perform well in class sitting.

Data augmentation had minor effects, yielding moderate F1 improvements, but didn't exceed the baseline CNN.

[5]The t-test used was from stats, and an independent t-test was performed https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html

Overall, pretrained and multimodal models show superior generalization and robustness, especially when working with limited or noisy data.

As a future step, we are willing to explore this classification by using a model such as vitpose-plus-small, to extract keypoint coordinates representing human joint positions. These keypoints will be encoded into structured feature vectors that capture the skeletal layout and movement cues. We will then train a lightweight classifier to perform the final classification into sitting, standing, or walking/running. This approach could improve robustness and interpretability, especially in scenarios with background noise or occlusions, by focusing on body structure rather than full-scene appearance.

References

Menglong Zhu Andrey Zhmoginov Liang-Chieh Chen Mark Sandler, Andrew Howard. 2019. Alternation. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

A Statistical Analysis of CNN vs FNN Performance

A t-test was conducted to compare the classification performance of CNN and FNN models across both original and background-removed datasets, using test accuracy scores obtained from five independent runs. On the original image dataset, the CNN achieved a mean test accuracy of 38.49% (± 5.04%), while the FNN obtained a lower mean accuracy of 28.97% (±3.74%). The resulting t-statistic was 3.3941 with a p-value of 0.0068, indicating that the difference in performance between CNN and FNN was statistically significant ($p < 0.05$), with CNN significantly outperforming FNN. In contrast, for the background-removed dataset, the CNN achieved a mean accuracy of 37.70% (±5.90%), and the FNN slightly outperformed it with a mean accuracy of 39.68% (±7.74%). However, the t-test yielded a t-statistic of -0.4561 and a p-value of 0.6581, suggesting that the difference was not statistically significant (p 0.05). This indicates that neither model consistently outperformed the other in the background-removed condition. Figure 5 shows a comparison of test accuracies over the runs.

B Strategy comparison in Transfer Learning

Explored Strategies

No Freezing

In this approach, all layers of the pre-trained MobileNetV2 model are left trainable. This allows the network to fully adapt to the new task but can lead to overfitting and longer training time, especially when the dataset is small.

Partial Freezing

All layers are initially frozen, then only the last few convolutional blocks are unfrozen. This provides a balance between retaining useful features from pre-training and adapting to the current task with fewer parameters being trained.

Feature Extraction

All pre-trained layers are frozen and only the final classifier is trained. This method is computationally efficient and helps prevent overfitting when the dataset is limited or similar to the original training data.

Feature extraction achieved the highest validation accuracy and exhibited a close alignment between training and validation loss, indicating reduced overfitting and better generalization. This is explored in the original dataset (Figure 6).

With background removed images, partial freezing resulted in the best validation accuracy, the training and validation losses had a larger gap, suggesting some overfitting. Feature extraction maintained more consistent loss trends but lower accuracy (Figure 8).

Based on performance across strategies, feature extraction was selected as the final strategy due to its strong validation accuracy and balanced training dynamics. Training time plots (Figures 7 and 8) also suggests that it is a comparatively good selection as it has lower training time, making it easy and feasible for the experiments. Loss and accuracy plots also support its stability, making it a suitable choice for final testing, which is confirmed by the final testing plots (Figure 10).

The report contains 2100 words excluding titles.

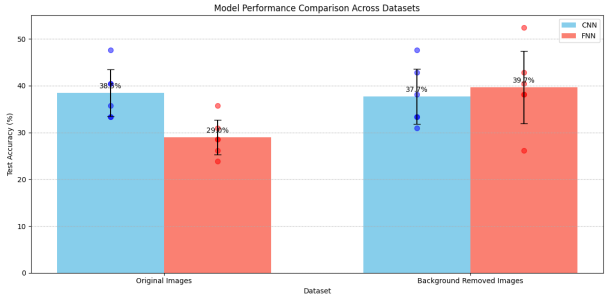

Figure 5: Comparison of test accuracies over the runs for CNN and FNN.

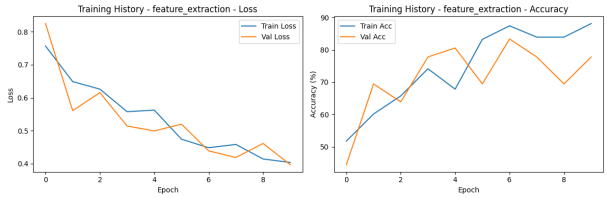

Figure 6: Loss and Accuracy variation in training and validation data with feature extraction strategy for original data.

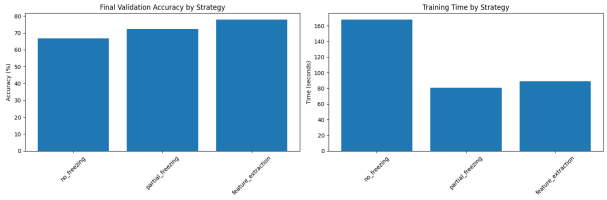

Figure 7: Final validation Accuracies and Training time comparison in each strategy explored for original data.

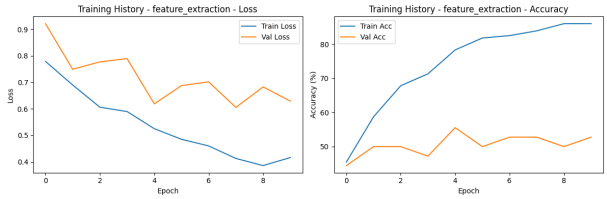

Figure 8: Loss and Accuracy variation in training and validation data with feature extraction strategy for background removed data.

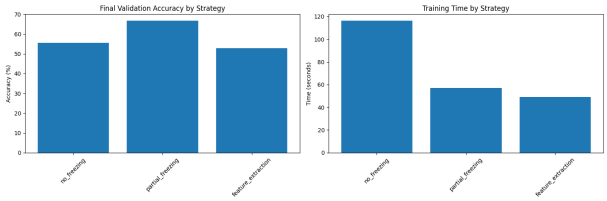

Figure 9: Final validation Accuracies and Training time comparison in each strategy explored for background removed data.

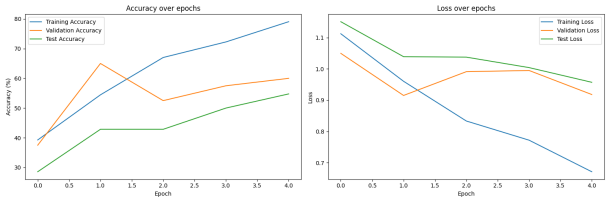

Figure 10: Loss and Accuracy variation in training, validation and testing data with feature extraction strategy for original data.