**IN3060/INM460 Computer Vision Coursework report**

* **Student name, ID and cohort:** Ismail Masanja (200039322) - PG
* **Google Drive folder:** <https://drive.google.com/drive/folders/1mGlfPHUqoTPRyyhMayi7z9rjytnI2EvD?usp=sharing>

# Data

Data is in CW\_Dataset folder divided into test and train folders. In each of these folders – train and test folders, there are images and labels folders. Images folders contain **different sized** **images** and labels folders containing text files having a numeric label representation of the respective image where, 0 = No Face Mask, 1 = Proper Wear (Face mask worn properly), 2 = Improper Wear (Face mask NOT worn properly).

Upon inspecting the number of items in train and test folders, we observe training set size = 2394 and testing set size = 458. The distribution is, For training set: Proper Wear = 1940, No Face Mask = 376 and Improper Wear = 78. For testing set: Proper Wear = 388, No Face Mask = 51 and Improper Wear = 19. This is an imbalanced distribution. Machine learning models perform best in normalized, balanced data.

To get good Training, validation and test datasets, the following is done: (i)Resize images to get uniform image size, (ii) Calculate channel mean and standard deviation of training set to normalize the data. (iii) Augment training set by adding some noise and flipping images to increase training size. (iv) Balance the classes of training and test datasets by Oversampling – to get uniform distribution. (v) Split training dataset into training and validation dataset. Effects are as of following figures:

|  |  |
| --- | --- |
| A blurry image of a person wearing a mask  Description automatically generated(a) Raw 64x64 images before normalization | A close up of a person's face  Description automatically generated  (b) After normalization, pictures are clearer |

# Top: Original distribution of Training and Testing datasets, Bottom: After steps (i) to (v) hence, getting more samples of balanced training, validation, and testing datasets.

|  |  |
| --- | --- |
|  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

# Implemented methods.

Three models are implemented, these are: (i) Scale-Invariant Feature Transform (SIFT) + Linear Support Vector Machine (SVM). (ii) A simple Convolution Neural Network (CNN) with Visual Geometry Group (VGG) architecture (SimpleVGGCNN). And (iii) Pretrained Mobilenet version 3, smaller version.

The Scale-Invariant Feature Transform (SIFT) + Support Vector Machine (SVM) model involves three stages. (i) Feature extraction – uses SIFT for extracting distinctive key points and descriptors from images that are invariant to image scaling, translation, and rotation. Example of image descriptors:

A group of colorful circles

Description automatically generated

(ii) Visual Bag of Words – Applies k-means clustering to the descriptors to create a visual bag of words, quantizing feature space into a finite number of categories which serve as input to the SVM. Number of clusters (***k***), k is 10 times the descriptor labels (factor of 10 widely is used) (iii) SVM is used for multiclass classification. Since, pytorch has no SVM class we create a custom one that also captures the k-means model as a parameter. To train this model we use hinge loss and Adaptive Moment Estimation (Adam) with learning rate scheduling for optimizing the network weights.

The Visual Geometry Group Convolution Neural Network (VGG CNN) – originally introduced by Fang et al (2017). In our implementation we use a simpler version using only 2 convolution blocks i.e. input block and hidden block, we also change ReLU layers to PReLU (adaptive learning of coefficient of leakage). For full architecture diagram see model\_training.ipynb. To train this model we use cross-entropy loss and Adaptive Moment Estimation (Adam) with learning rate scheduling for optimizing the network weights.

MobileNet architectures are designed for mobile devices, providing a good balance between speed and accuracy. Due to limited compute resources, we use pretrained mobilenet version 3 small (mobilenet\_v3\_small), for full architecture diagram see model\_training.ipynb.. To fine-tune the model, we replace the final classification layer to suit the number of target classes (3 classes). To train the model fit our data, we use cross-entropy loss and Adaptive Moment Estimation (Adam) with learning rate scheduling for optimizing the network weights.

For evaluation and performance metrics, the following methods are implemented. (i) Model evaluation class includes metrics such as accuracy, precision, recall, and F1 score, class classification report from sklearn and confusion matrix to assess model performance. (ii) Random sampling visualization class – displays random samples of predictions to provide visual feedback on the model’s performance. (iii) Data distribution visualization class – plots the distribution of class labels in the training, validation, and testing sets to evaluate the balance of data and its potential impact on model training. For full code, refer to *code folder* in the google drive link provided in this document first page head.

# Results

Precision is a measure of how close repeated measurements of the same quantity are to each other. It is often used in the context of scientific experiments to assess the reliability of the results. Recall is a measure of how many relevant items are retrieved by a search or classification algorithm, relative to the total number of relevant items. It is often used in information retrieval and machine learning to assess the performance of a system in retrieving relevant information. The F1 score is a measure of a test’s accuracy that considers both precision and recall. It is the harmonic mean of precision and recall and is used to evaluate the performance of binary classification systems.

In summary, SIFT + Linear SVM model trained for 45 minutes – 15 epochs on CPU (most computations happen on CPU hence, changing to GPU does not cut time). It has an accuracy of 46.56%, precision of 46.38%, recall of 46.56% and f1-score of 46.15% which is slightly better than 1/3 no of classes random draw probability. Due to the simplicity of the model relative to the image data, The SIFT + SVM model shows moderate effectiveness, with accuracy and other metrics around 46%. This suggests that while the model can identify features and classify images, its performance is limited.

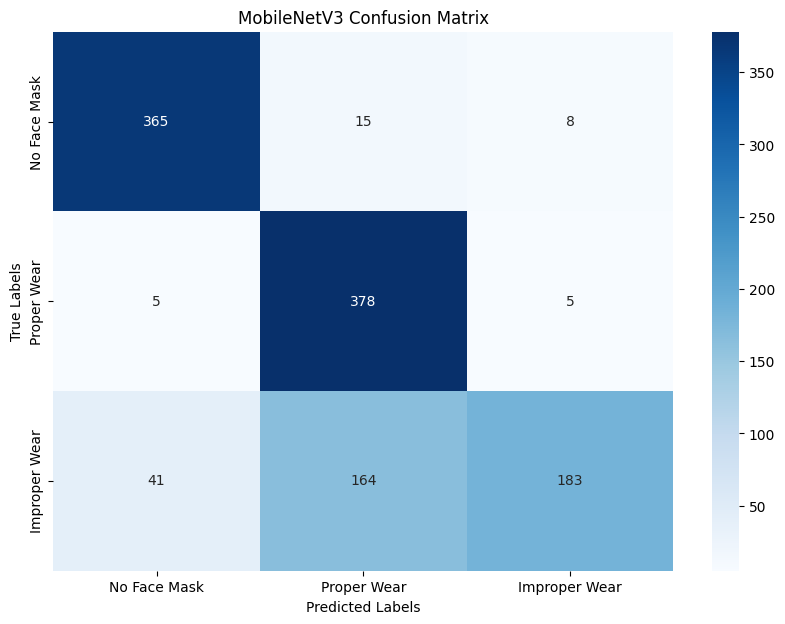
Simple VGG CNN model trained for 15 minutes – 20 epochs on a GPU. It has an accuracy of 77.32%, precision of 80.75%, recall of 77.32% and f1-score of 75.10%. This model significantly outperforms the SIFT + SVM model, demonstrating a much higher accuracy and other metrics. The substantial improvement highlights the advantages of using deep learning and convolutional neural networks for image-based tasks, especially given their ability to learn hierarchical features.

Pretrained mobilenet model trained for 13 minutes – 20 epochs on a GPU. It has an accuracy of 79.55%, precision of 83.35%, recall of 79.55% and f1-score of 78.01%. This model also performs well, slightly better than the Simple VGG CNN. The efficiency of Mobilenet, which is designed for mobile and low-power devices, might have resulted in a slight trade-off in performance bigger versions might perform even better.

The confusion matrices for these 3 models are as follows:

|  |  |
| --- | --- |
|  |  |

(a) SIFT + SVM confusion matrix (b) Simple VGG CNN confusion matrix



(c) Mobilenet V3 small confusion matrix

From figures above, (a) SIFT + SVM model shows high accuracy in predicting Proper wear (219) however, accuracy really drops on improper wear, whose column is evenly scattered between the 3 classes hence, improper wear still not learned. (b) Simple VGG CNN has higher accuracy in predicting Proper Wear (372) and No Face Mask (367) however, accuracy drops between proper wear and Improper Wear. (c) Mobilenet V3 small is slightly better in predicting Proper wear (378) and No Face Mask (365) than (b) but still, there is an issue between proper wear and improper wear. Referencing from the data section, original data did not have many Improper wear sample hence we need more. In conclusion **we select Mobilenet V3**.