CS4243 - Computer Vision and Pattern Recognition - Mini Project

Threat Image Classifier

Group 30

S Dinesh Raj | Myat Htet Kyaw | Yong Ray Wen Joshua | Muhammad Hanif Bin Muhammad Kamal

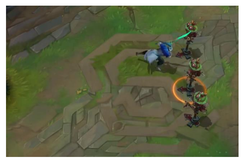
Dataset (cs4243_smallest)

Still images from Dataset; No videos were processed.
No of Classes: 3

Class: **Normal**	Class: **Carrying**	Class: **Threat**
Sample: **1857**	Sample: **1535**	Sample: **1551**

Data Preprocessing

1. Manual relabelling and removal of pollution



"Threat" → removed

"Carrying" → Threat

Class	Normal	Carrying	Threat
Count	1567	1021	540

2. Manual undersampling to remove imbalance

Preferentially remove similar images

Class	Normal	Carrying	Threat
Count	500	500	500

Baseline CNN

1. Baseline CNN implementation

Resize each image to **100x100** pixels
Converted the images to **grayscale**
Randomised selection from **unfiltered** dataset

5 convolutional layers, 2 dense layers
Epochs = 10
Actionativation function = relu(for all layers), softmax (final)
Optimiser = **Adam**

Training Accuracy**: 66%**
Validation Accuracy: **38%**

1. CNN implementation with cleaned dataset

Hyperparameter tuning
- Changed learning rate
- **Higher** epochs

Training Accuracy: **86%**
Validation Accuracy: **45%**

Observed: Overfitting

Combating Overfitting in CNN

1. Data Augmentation

Introduce image transformations as a preprocessing layer in model

Transformations include:
- Random zoom, horizontal and vertical flip, rotation

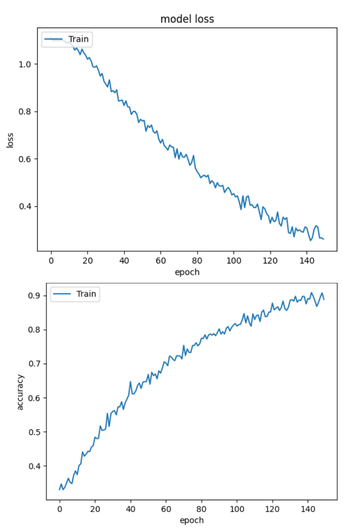
Effectively **increase** size of **training dataset**
Training Accuracy: **76%**
Validation Accuracy: **59%**

1. Add dropout after each layer

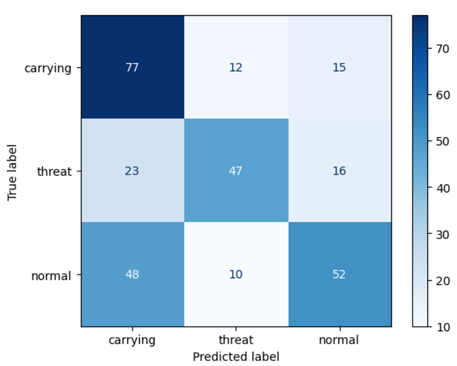
Add dropout in all layers incremental in levels of 20% to 40%.
Training accuracy**: 96%**
Validation Accuracy: **65%**

Conclusion:

Despite increasing model accuracy and minimising overfit interval, we were able to reduce overfit but not eliminate it.



F1 Score: 0.585474275880426
Precision: 0.6093119048855037
Recall: 0.5865411720062883
(Macro Averages)



TRANSFER LEARNING - ResNet18

Pre-Trained Neural Network.
18 layers with more than 1,000,000 images.

Changed the final layer to 3 (to represent classes).
Training Accuracy: **94%**
Validation Accuracy: **75%**

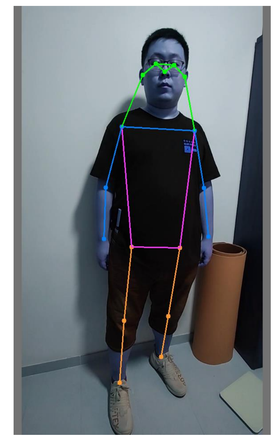
In efforts to reduce overfit, added dropout of 0.2 to the final layer.

Training Accuracy: **80%**
Validation Accuracy: **78%**

Object Detection Techniques

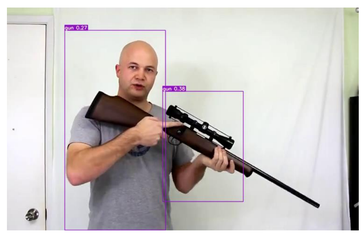
Pose Detection with YOLOv7

Processed dataset using YOLOv7 with pretrained model weights and used this new dataset with poses highlighted as **training dataset** for our **improved CNN** but no significant improvement in validation accuracy observed.



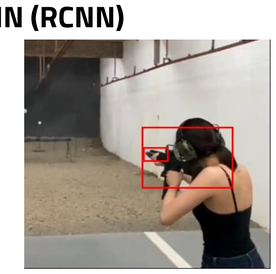
Object Detection with YOLOv7

Trained model on dataset of >8000 images with YOLOv7 labels
Poor performance, likely due to insufficient epochs.



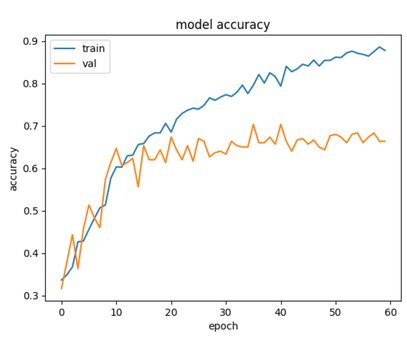
Object Detection with Region-based CNN (RCNN)

Using model with pre-trained weights on a dataset of only 333 images of pistols and rifles. Good performance in detection without labelling.



Using Object Detection to Crop Images

Using saved points from both **pose detection** and **weapon detection** to determine a new boundary to extract out the most important features and ignoring the surrounding environment.



Training CNN with previous enhancements on new dataset, accuracy is slightly increased.

Training Accuracy: **88%**
Validation Accuracy: **68%**

MEMBER CONTRIBUTION

1. S Dinesh Raj
   - Data preprocessing, Baseline CNN + fine tuning (Dropout), Resnet
1. Myat Htet Kyaw
   - Data preprocessing, Pose detection, Object detection
1. Yong Ray Wen Joshua
   - Data preprocessing, Object detection, Further preprocessing with object detection
1. Muhammad Hanif Bin Muhammad Kamal
   - Data preprocessing, Data augmentation implementation in CNN