CS4243 Image Classification for Weapon Detection

Group 27 Not Overfitting

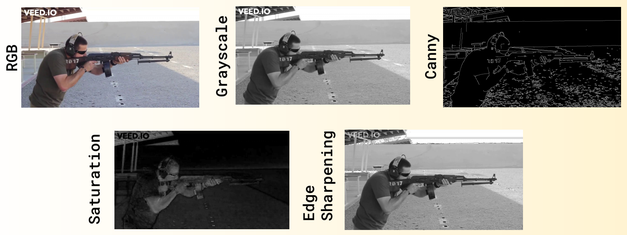
Team Members: Heinrich (A0206403W), Anthony Lie (A0200691N), Juan Davin Lie (A0201229U), Liu Ernest Hin Yui (A0216284E)

Stage 1: Preprocessing Trials

At the beginning of the project, we decided to perform several tests to determine which preprocessing method is the best for our purpose.

We decided to train different models for different types of images. Below are examples of the different types of images we trained different models with.

We hypothesized that whilst the model being fed original RGB images would perform the best as it retains all colour information, the drop in accuracy for other types of images such as grayscale or saturation will not be too drastic. Thus, using other image types may be sufficient for our purpose in order to lower training time as our group members lack the computational power to conduct tests on very complex models.



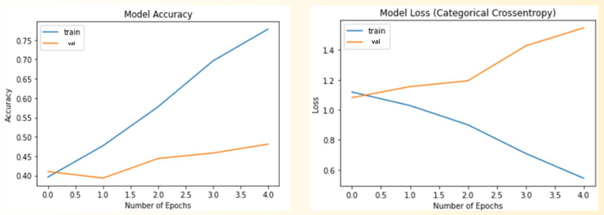
Grayscale: Testing to see if colour information is necessary since weapons are mostly black and white
Saturation: Weapons generally have very low saturation
Canny: Testing importance of edge information only
Edge sharpening: Testing importance of edge information on grayscale image

To better allow our model to generalize to different kinds of images, all preprocessing tests were also performed with several augmentation steps on our training data such as:

- Random horizontal and vertical shifts
- Random zoom
- Random shear shifts (max shear range of 15 degrees)
- Random horizontal flips
- Random rotation (max rotation range of 15 degrees)
- Noise
- Blur
- Noise and blur

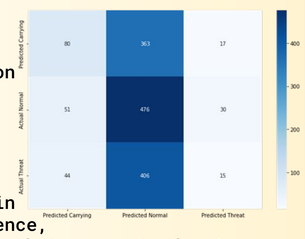
Best Accuracy for each preprocessing type:
RGB (with Noise and Blur and other augmentation): 48.11%
Grayscale (No augmentation): 46.15%
Saturation (Same across all): 37.56%
Edge Sharpening (No Augmentation): 44.67%
Canny (No Augmentation): 40.55%

RGB Accuracy and Loss Graph and Confusion Matrix
Model Used: LeNet-5



Stage Evaluation:

By looking at the training curves, we see that validation accuracy and loss are both increasing. Thus, despite the increasing accuracy, the model is becoming increasingly uncertain when classifying the images in the validation dataset and hence, a loss increase. This means that amongst the images that are correctly predicted by the model, these predictions are "borderline predictions", in that the model is very unconfident when classifying these images.

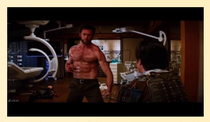


A quick inspection of the confusion matrix shows that the model is simply giving a classification of "Normal" for most images. This could be because a majority of the training images come from the "Normal" class or in other words, an imbalanced dataset.

Stage 2: Dataset Cleaning

To better understand the dataset, we manually looked through the images and found quite a number of invalid data. We decided to remove images from each category using the following criteria:

- Threat Images Removal Criteria:
  - No persons
  - "Toy" weapons
  - Weapons other than knives or guns
  - Hard to see weapons
  - Person is not looking at where the gun is aiming
  - Person is not holding a gun or a knife


Person is not holding a gun or a knife


Person is not looking at where the gun is aiming

- Carrying Images Removal Criteria:
  - No persons
  - "Toy" weapons
  - Weapons other than knives or guns
  - Hard to see weapons
  - Person is looking at where the gun is aiming
  - Person is not holding a gun or a knife


Hard to see weapons


No person

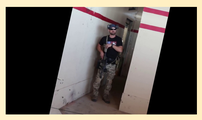
- Normal Images Removal Criteria:
  - No persons

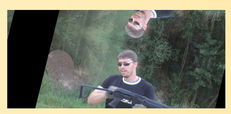

No person


No person

Dataset Image Augmentation

Upon further investigation, we discovered that the raw dataset size is not balanced across the three categories (normal > carrying > threat). This could lead to biased training result as depicted in Stage 1. To balance the dataset, we performed upsampling on the minority classes using augmentation techniques (rotation, shift, reflection, shear, reflect fill mode), and downsampling for the class with the most images. We ended up with a balanced dataset of 1800 images per class. The following are some examples of augmented images we created:





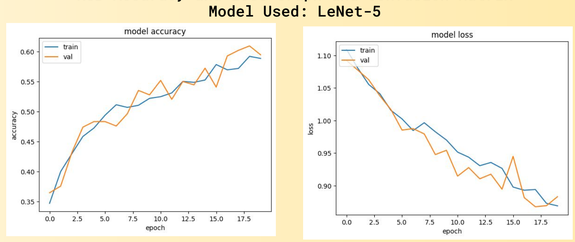
Dataset Distribution

After balancing the dataset, we split the dataset into 3 train, validation and test with split ratio of 80%,10%,10% respectively.

Post-Cleaning Results

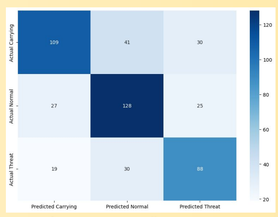
Test Accuracy: 65.39%
Recall Macro: 65.30%

RGB Accuracy and Loss Graph and Confusion Matrix
Model Used: LeNet-5



Stage Evaluation:

By looking at the training curves, we see that validation accuracy is now generally increasing with the training accuracy, and validation loss is generally decreasing with the training loss. Thus, the poor results attained in the previous stage could be due to the dirty dataset. "Garbage in, garbage out."



Previously, the model was classifying almost everything as "Normal". At the end of this stage, the confusion matrix shows us that the model is now able to classify most of the images in each class.

Stage 3: Model Diagnostics

Baseline LeNet-5 Diagnostics

We tried using the most basic LeNet-5 model for our baseline because it saves a lot of computational time and it paints a rough picture on the effectiveness of our data cleaning and the different preprocessing steps.

For the architecture of the model, refer to **Figure 1.** at the bottom. We trained this model with learning rate of $1e^{-3}$, using the Adam optimizer with an Early Stopping patience of 5 epochs to avoid overfitting.

Enhanced LeNet-5:

Upon discovery of the model's still perform poorly, we hypothesized that it was due to a lack of convolutional feature extractors.

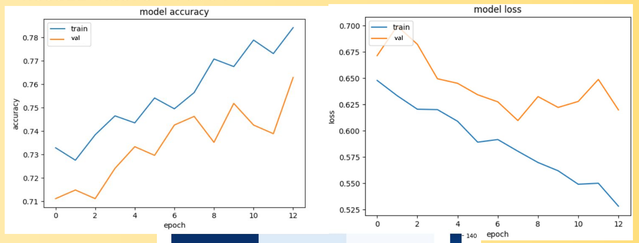
As a result, we decided to enhance the baseline model by adding more convolutional layers to it. To reduce the chances of overfitting, we further performed regularization by adding an additional dropout layer with a rate of 0.2.

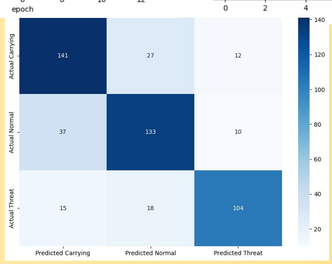
The new additional layers are shown in **Figure 2.** below. We trained our dataset with this model with learning rate of $1e^{-4}$, using Adam optimizer with an Early Stopping patience of 5 epochs.

These enhancements were inspired by the Inception V3 model, which was pre-trained by ImageNet, when we were experimenting it using the same processed dataset.

RGB Accuracy + Loss Graph + Confusion Matrix after data cleaning

Test Accuracy: 76.05%
Recall Macro: 76.04%





Stage Evaluation:

By looking at the training curves, we can see that the validation loss generally decreasing and always higher than the training loss, while the validation accuracy is generally increasing.

The minimum loss attained is also lower than that of the previous stage, at around 0.620 compared to 0.820 before. This tells us that the model, whilst now able to make more accurate predictions, is also more confident than before, which explains the generally lower loss.

The confusion matrix shows that the enhanced model is able to attain higher accuracy scores for every class as compared to stage 2. As a result, the prediction accuracy per class is higher than before, at about 77% on average.

Team Contribution

Stage 1: Preprocessing Trials
- RGB: Heinrich, Anthony, Juan, Ernest
- Grayscale: Ernest
- Saturation: Juan
- Edge sharpening: Anthony
- Canny: Heinrich

Stage 2: Dataset Cleaning
- "Normal" class: Juan
- "Carrying" class: Anthony
- "Threat" class: Ernest
- Image Augmentation & Dataset Split: Heinrich

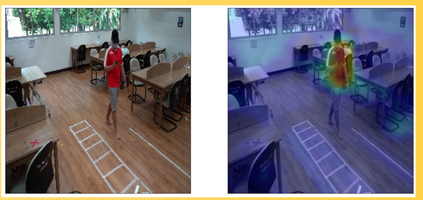
Stage 3: Model Improvement:
- Model Diagnostics: Ernest
- Enhancing Model: Heinrich

Stage 4: Model Interpretability
- Grad-cam: Anthony

Stage 4: Model Interpretability

Neural networks are notorious for their lack of interpretability, which understandingly, instils doubts in the model. To circumvent this issue, our group has decided to use the Grad-CAM technique to gain some insight into how our model makes its decisions.

The following are a few examples of images which Enhanced LeNet-5 has correctly, and wrongly classified:


Actual Normal, Predicting Normal


Actual Carrying, Predicting Carrying


Actual Threat, Predicting Threat


Actual Normal, Predicting Carrying


Actual Carrying, Predicting Normal

Actual Threat, Predicting Normal

From the Grad-CAM plots, we observed that the model tends to focus in on the hands and head of the person in the image, even for incorrect classifications. When there is a weapon in the image, the model is also largely able to focus on that, as depicted in the second plot.

That being said, the model still fails at recognizing weapons at awkward angles. Take the sixth plot for example, where the woman is pointing the gun almost towards the camera. Despite the model being able to detect the woman's arms, it is unable to tell apart the gun from the woman's body. Take a look at the fifth plot as well, where half of the knife is cut off from the image. In such cases where the weapon is not fully captured, the model also fails to classify correctly.

Figure 1. Baseline Lenet 5 Architecture

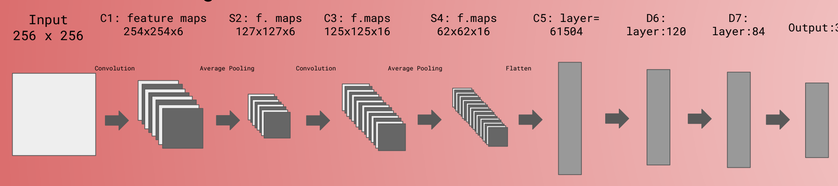
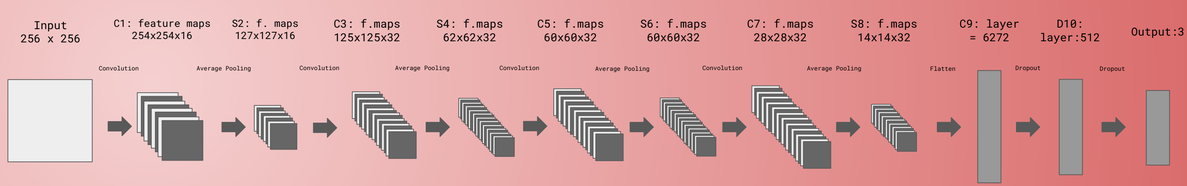


Figure 2.Enhanced Lenet 5 Architecture



Further Improvements

- Pretrained our model to detect weapons by feeding it with images of weapons.
- Much deeper and complex models with more convolutional layers like InceptionV3.
- More trial and error to find the best hyperparameter.