Report by: Scharmensey Fine

Detecting Images Containing Weapons with Deep Learning

Introduction

Technology has improved significantly throughout the decade. This can be seen as a positive milestone, as advancements in medicine allow for limb replacement with prosthetics or even laptops being offered in schools. However, there are also the downsides to the advancements in technology, but for research purposes we will be focusing on one. Which are the negative impacts that come with advancement in weaponry. Guns have been extremely normalized in the US, where the number of school shootings are increasing yearly. It is important to install security measures in almost every location to look after individuals. In today's research we will be aiding in this issue, and the objective is to use machine learning to classify images that contain a weapon.

Discussion

This classification will be done by constructing a Convolutional Neural Network to learn the pattern & structure of the dataset. The dataset being used is built up of unique images from stock vault, where image selection was crucial in getting proper results. The images were placed into two class folders (weapon or safe). The weapon category contains 382 images corresponding to either people with guns & knives or the weapon itself. The safe category contains 300 images that don't correspond to weapons at all, only non-threating objects/people in safe settings holding non-threating items. For the deep learning to run efficiently, you must provide & feed the network with extensive amounts of images.

But why a Convolutional Neural Network out of all neural networks? Well Convolutional Neural Networks (CNNs) rank high amongst other machine learning algorithms in handling complex imagery. This is due to its convolutional layers, they can capture low-level features and progressively learn higher-level features like textures, patterns, and object parts. This hierarchical feature extraction is helpful to have for image comprehension. Also, the network's pooling layers reduces the spatial dimensions of the feature maps while retaining the important information. This leads to a form of spatial hierarchy, where higher-level features become more abstract and robust to spatial translations. CNNs also offer allot of flexibility which is always great. But overall, the goal is to build a proficient Convolutional Neural Network to detect weapons in images with an accuracy above 85%.

Images from the dataset loaded into the neural network with its corresponding label

Application

Can be seen on the Detecting Weapons.py file….

Results

After training and tuning the model, the network was utilized in classify weapon appearance on the testing data which yielded an accuracy of 69%. This is almost 15% below our goal and can be greater. Looking the classification report on the right, the "0" represents images with weapons, and the "1" represents images without weapons (safe). The model had an F1-score, precision, & recall weighted average around 0.69. The recall is 0.54 for the weapon class, indicating this model has a tougher time classifying images with weapons correctly. You can even see this with the 0.69 precision compared to the 0.81 recall in the non-weapon class, more of the weapon images are getting misclassified compared to the images that are considered "safe" without a weapon.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.54      0.61        74
           1       0.69      0.81      0.74        93

    accuracy                           0.69       167
   macro avg       0.69      0.67      0.67       167
weighted avg       0.69      0.69      0.68       167
```

The network is having trouble recognizing the guns/knives in images, and when it's not found then classifies them wrong. The training process reflects this, the model learns well with the training data as the losses consistently decrease over each epoch *(exhibit 3)*, however this shows that more can be learned and implemented on the test data. And looking at the ROC curve *(exhibit 4)* there's a clear performance difference between the training and testing set, to where the model is over fitting the data. Dropout regularization & a learning rate scheduler were added, however the overfitting still occurred.

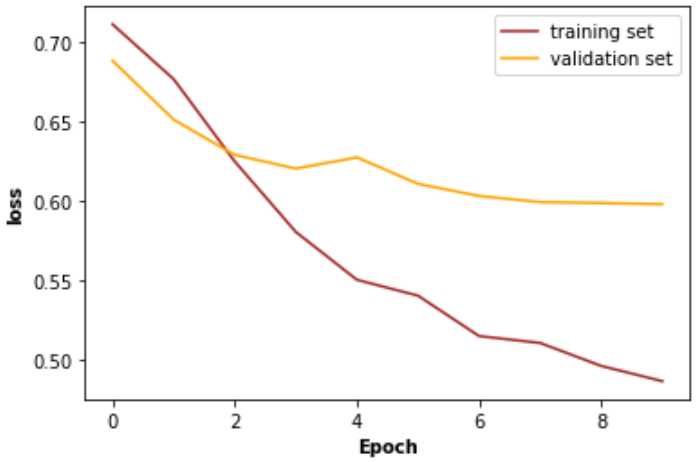

Exhibit 3

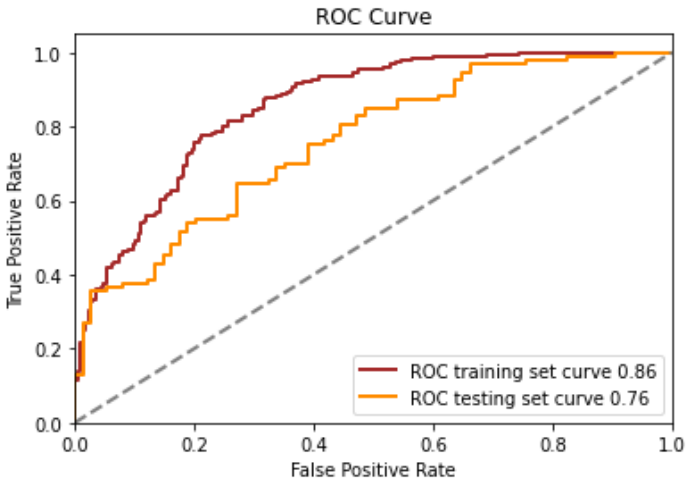

Exhibit 4

In the data set, not all the images with guns are super noticeable. There are a few images where weapons are being held by people, getting covered by their hands, so this takes extra effort to identify those weapons. The model is having some trouble isolating weapons in those situations.

Conclusion

Altogether, we were able to build a convolutional neural network to classify images that contain either a knife or a gun. After many modifications and tunning, the model was able to perform with an accuracy of 69% at best but ranged in the high 60s. This isn't super satisfactory but is more realistic, aiming for an 85% would require an extensive amount of photos/storage in order for the model to learn properly. There were a lot of takeaways from this project, being image selection & quantity are significant in image classification, being imperial for the Neural Network to learn. Also making sure images are clearly showcasing what you want the network to learn is key. Understanding the architecture of the neural network is another tool that plays a crucial part of deep learning. To know how to adjust parameters, you need to know how your model is working, how many/types of layers being used, the size of each layer, it's all the center of the neural network. The training part stems from how well the architecture is. And going back to how image selection is important, your architecture should be built/based on your images. Architecture of the network & quality of datasets are both what needs to be prioritized. After making improvements on those two things, a working CNN in detecting weaponry within images was obtained.

References

- Images derived from Stock Vault
- Image preparation: https://github.com/nicknochnack/ImageClassification/blob/main/Getting%20Started.ipynb – *nicknochnack*
- Dive into Deep Learning – *ASTON ZHANG, ZACHARY C. LIPTON, MU LI, AND ALEXANDER J. SMOLA*