Image Detection using CNN: Contraband Classification

Team: Gersian Collaku & Emilie Xiao

Agenda

01 Problem Statement

02 Methods

03 Challenge

04 Results

05 Conclusions

06 Fun Fact

01

Problem Statement

Problem Statement

What?

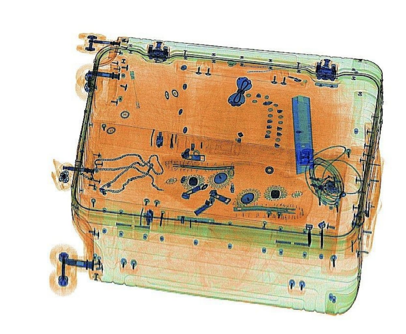
- Recognize the type of contraband in a luggage

Why?

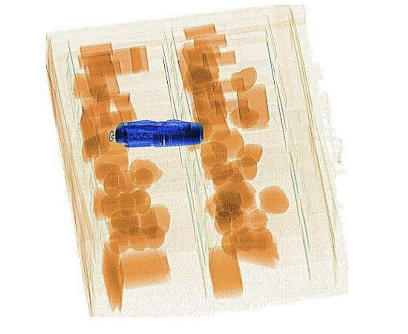
- Enhance both security (detect hard-to-see items) and efficiency (lead to shorter lines)

X-ray Image Examples

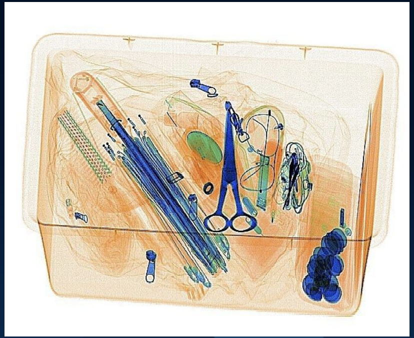
Utility Knife



Multi-tool Knife

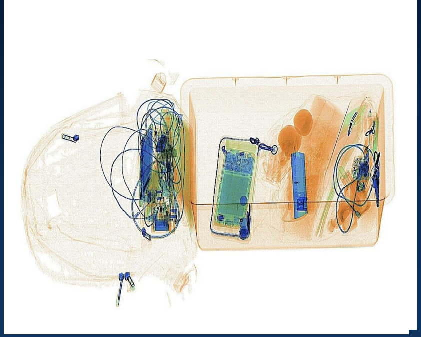


Scissors



Straight Knife

Folding Knife



02

Methods

Convolutional Neural Network (CNN)

Methods

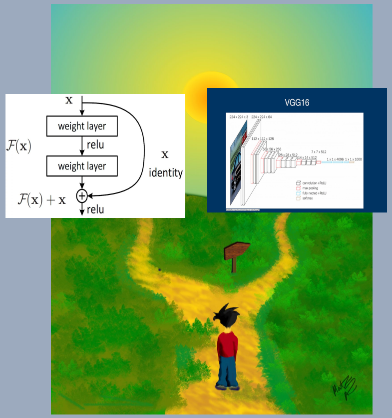
1. We failed…

2. Then we explored different options

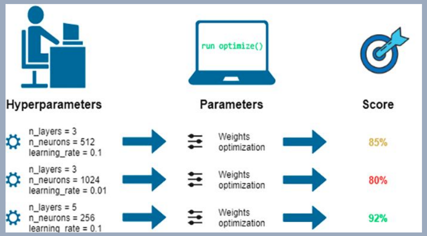
3. Hyperparameter Tuning and Data Augmentation

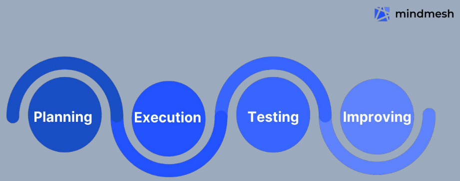
4. A lot of tests

```
Epoch 11/20
124/124 [============================] - 247s 2s/step - loss: 1.4069 - accuracy: 0.2879 - val_loss: 1.9056 - val_accuracy: 0.2332
Epoch 12/20
124/124 [============================] - 249s 2s/step - loss: 1.4093 - accuracy: 0.2728 - val_loss: 2.0957 - val_accuracy: 0.2332
Epoch 13/20
124/124 [============================] - 241s 2s/step - loss: 1.4066 - accuracy: 0.2839 - val_loss: 2.0118 - val_accuracy: 0.2311
Epoch 14/20
124/124 [============================] - 249s 2s/step - loss: 1.4057 - accuracy: 0.2877 - val_loss: 2.0179 - val_accuracy: 0.2311
Epoch 15/20
124/124 [============================] - 248s 2s/step - loss: 1.4063 - accuracy: 0.2861 - val_loss: 2.0415 - val_accuracy: 0.2311
Epoch 16/20
124/124 [============================] - 250s 2s/step - loss: 1.4068 - accuracy: 0.2882 - val_loss: 1.9349 - val_accuracy: 0.2311
Epoch 17/20
124/124 [============================] - 252s 2s/step - loss: 1.4060 - accuracy: 0.2745 - val_loss: 1.9207 - val_accuracy: 0.2311
Epoch 18/20
124/124 [============================] - 250s 2s/step - loss: 1.4046 - accuracy: 0.2831 - val_loss: 1.9695 - val_accuracy: 0.2332
Epoch 19/20
124/124 [============================] - 236s 2s/step - loss: 1.4057 - accuracy: 0.2776 - val_loss: 1.9698 - val_accuracy: 0.2254
Epoch 20/20
 97/124 [=====================>.......] - ETA: 44s - loss: 1.4066 - accuracy: 0.2623
```



Resnet vs. VGG?





03

Challenge

Challenge

- Imbalanced Dataset (amount of images in the Train, Validation, Test sets & the distribution of the contraband types throughout the images)

Solution:

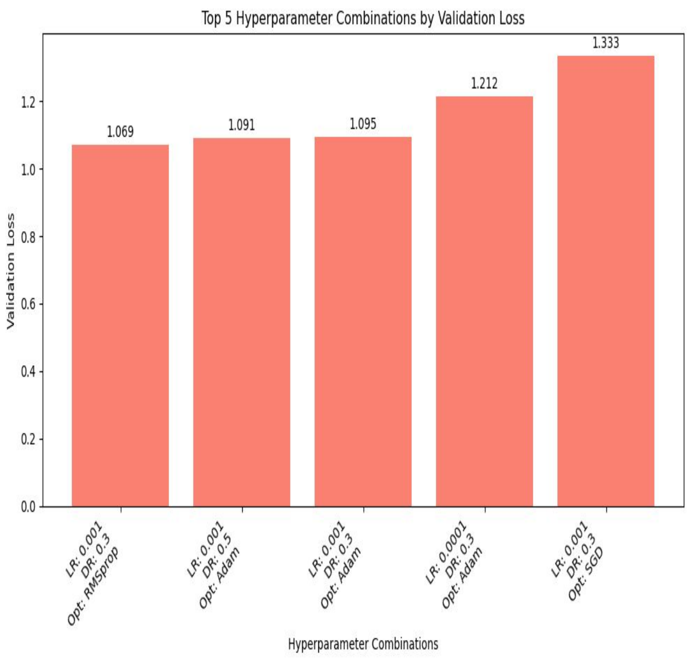
- Step 1 – **Consolidate Images**
- **Step 2 – Redistribute the Images**

- **Step 1 – Oversampling the Minority Class with Data Augmentation:** "Straight Knife"
- **Step 2 – Undersampling the Majority Class:** "Utility Knife"

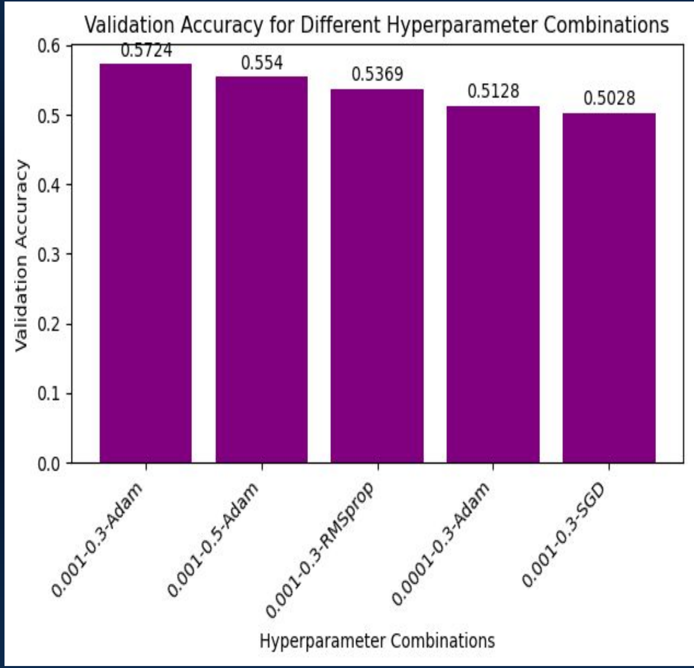
04

Results

Results

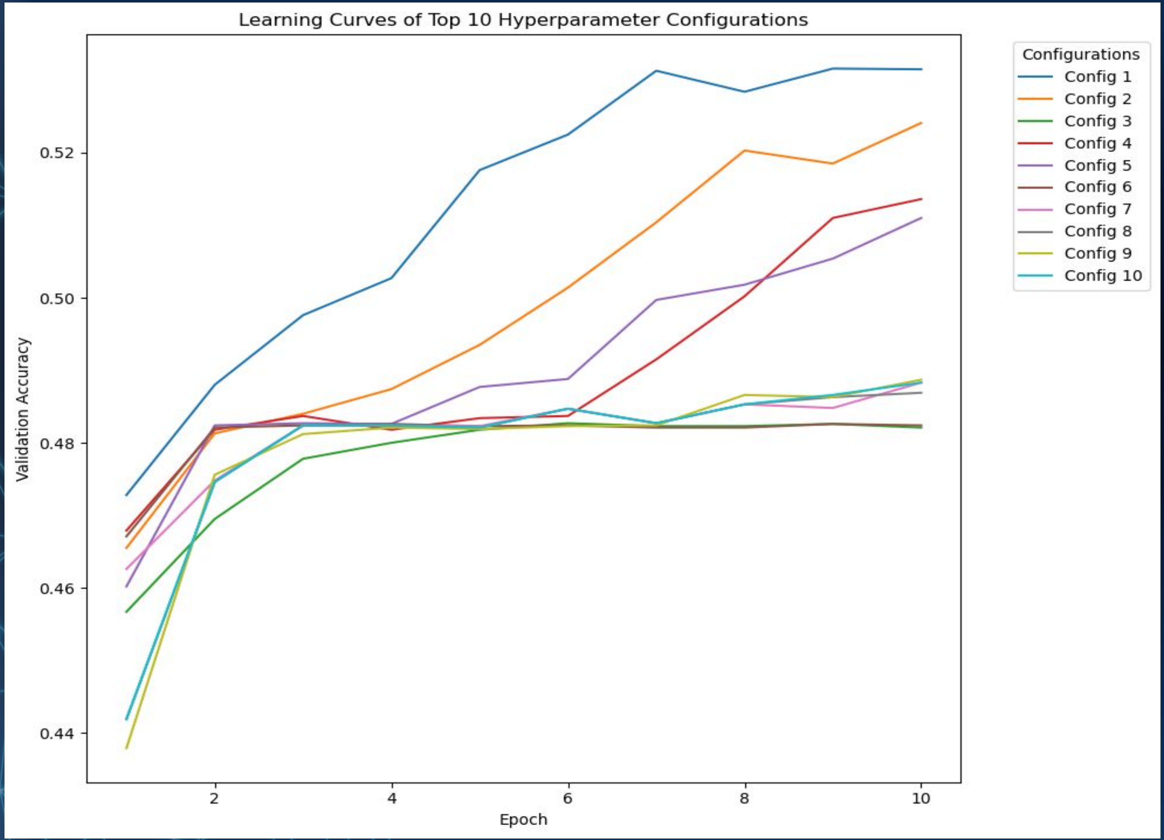


Validation Loss



Validation Accuracy

Results



KEY:

Config 1: Learning Rate: 0.001, Dropout Rate: 0.3, Optimizer: Adam

Config 2: Learning Rate: 0.001, Dropout Rate: 0.3, Optimizer: RMSprop

Config 3: Learning Rate: 0.001, Dropout Rate: 0.3, Optimizer: SGD

Config 4: Learning Rate: 0.001, Dropout Rate: 0.5, Optimizer: Adam

Config 5: Learning Rate: 0.0001, Dropout Rate: 0.3, Optimizer: Adam

Config 6: Learning Rate: 0.001, Dropout Rate: 0.5, Optimizer: RMSprop

Config 7: Learning Rate: 0.001, Dropout Rate: 0.7, Optimizer: Adam

Config 8: Learning Rate: 0.0001, Dropout Rate: 0.5, Optimizer: Adam

Config 9: Learning Rate: 0.0001, Dropout Rate: 0.3, Optimizer: RMSprop

Config 10: Learning Rate: 0.0001, Dropout Rate: 0.7, Optimizer: Adam

05

Conclusions

Conclusions

What we learned:

- The importance of a balanced dataset before running the model
- The crucial role hyperparameter tuning plays in model performance

06

Fun Fact

Fun Fact

Turns out, if the human eye struggles to make out what it's seeing in an x-ray image, a computer might just squint harder.

Thank You!