Detecting dangerous items in X-Ray luggage scans

Enhancing travel security is crucial

Security when travelling has always been a major concern for both commercial and private aviation. X-Ray scans have played an important role in securing passenger safety, and the use of machine learning in this field is growing rapidly

Complexity of images is a challenge

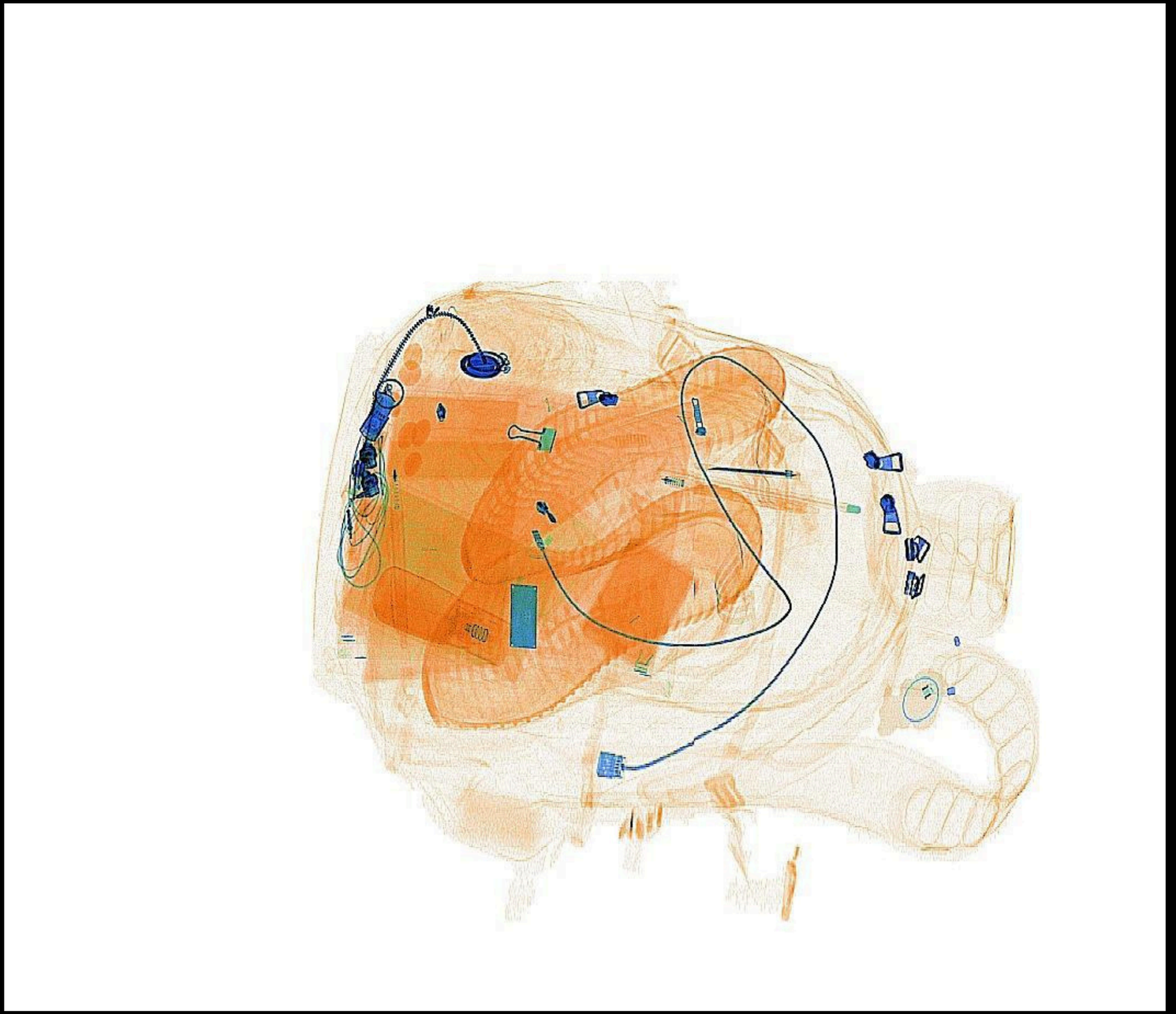
Complexity of images is a challenge

Sometimes it can be hard to spot a dangerous item in a luggage. These tools can help security professionals and could manage to reduce their workload.

Complexity of images is a challenge

Sometimes it can be hard to spot a dangerous item in a luggage. These tools can help security professionals and could manage to reduce their workload.

Can you spot the knife?



Dangerous items

The model is trained to recognize the presence of eight different classes of dangerous (or potentially dangerous) objects:

- Cameras
- Cellphones
- General electronic devices
- Knives
- Laptops
- Lighters
- Powerbanks
- Scissors

Augmentation

The dataset was augmented by applying random rotations and random cropping to images. This allowed classes to have a seemingly uniform distribution over the dataset

Augmentation

The dataset was augmented by applying random rotations and random cropping to images. This allowed classes to have a seemingly uniform distribution over the dataset

Augmentation

The dataset was augmented by applying random rotations and random cropping to images. This allowed classes to have a seemingly uniform distribution over the dataset

The Model

The Model

The model is a CNN with two convolutions followed by batch normalization and max pooling, and two fully connected linear layers

The Model

The model is a CNN with two convolutions followed by batch normalization and max pooling, and two fully connected linear layers

gaussianblur -> edgedetection -> conv1 -> batchnorm -> tanh -> maxpool2d -> conv2 -> batchnorm -> tanh -> maxpool2d -> linear -> tanh -> linear

Pipeline

Pipeline

Developed entirely with PyTorch

Pipeline

Developed entirely with PyTorch

During the initial transformations, images are resized to 64x64 and converted into PyTorch tensors

Pipeline

Developed entirely with PyTorch

During the initial transformations, images are resized to 64x64 and converted into PyTorch tensors

Normalization is performed with respect to each class

Pipeline

Developed entirely with PyTorch

During the initial transformations, images are resized to 64x64 and converted into PyTorch tensors

Normalization is performed with respect to each class

SGD is used over Cross Entropy Loss in coonjunction with L2 regularization

Performance

Performance

Performance was evaluated on the validation set after around 60 epochs of training.

Performance

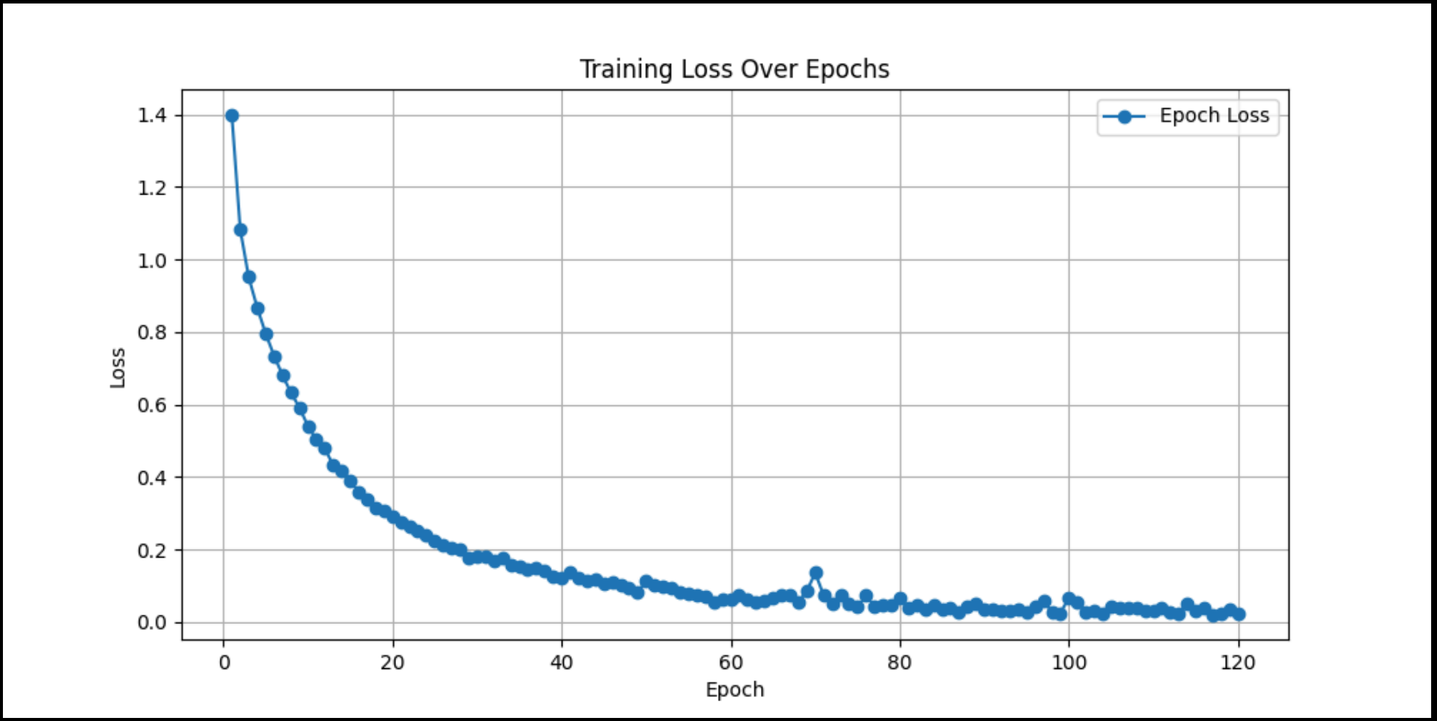
Performance was evaluated on the validation set after around 60 epochs of training.

Through trial and error it was possible to reach over 90% of accuracy on multiple validation sets for all classes

Performance

Performance was evaluated on the validation set after around 60 epochs of training.

Through trial and error it was possible to reach over 90% of accuracy on multiple validation sets for all classes



Project code at

https://github.com/ricvigi/ailab_project_private