THE UNIVERSITY OF SYDNEY

Automated Baggage Screening System with Deep Learning Neural Networks

QBUS3820 Machine Learning and Data Mining in Business

Group 1

Full Name	Student ID
Charles Wang	520033177
Hugh Qiu	520005325
Shaw Wang	520570177
Martin Pua	510053846

Tutor: Chengbo Wang

Sydney, May 23, 2024

Contents

1 | Task

X-ray imaging has been widely deployed in securing strategic infrastructures and properties by uncovering hidden objects within visitor's baggage. In the field of security, the accuracy of detecting a prohibited item in baggage using X-ray imaging is essential but has always been a challenge. Manual detection of prohibited items has not been very efficient especially during peak hours as the inspector needs strong focus recognising prohibited items.

Therefore, demand has rose for the use of automated security inspection system for effective processing of inspections, and it is becoming a trend. This system is required to have the function of alerting inspector of any potential prohibited items with high accuracy, as the consequences can be risk of a catastrophic incident. Sensitivity needs be limited too as alerting the inspector for everything is not ideal. Since X-ray passes through objects and display the object's contour, but object's surface may not be shown. Moreover, items in baggage are often rotated and overlapped with each other. These issues have caused challenges for the system to recognise prohibited items. A deep learning model is therefore suitable, great in dealing with unstructured data using computer vision and make decisions.

2 | Dataset

A public dataset recently published from Ma et al. (2024) was used, the data contains 32,000 X-ray images (16,000 pairs, each pair contains an orthogonal view and a side view image) of baggage, with 10,000 images are (5,000 pairs) positive sample (contains prohibited items). The data contains 15 labels of prohibited items that is illegal in most subway stations. It is important to be noticed that some sample image (pair) may contain multiple labels (multiple prohibited) items, for example, positive sample 1 have two labels (`Battery` and `Dart`). Figure 2.1 shows some examples of paired X-ray images of the dataset.

To summarise, the database of security checks dual-view X-ray images have the following characteristics:

- 3 channels (RGB) of each view
- 600*800 pixels of each image
- 2 views: orthogonal and side
- overlapping items in the images, and the items may be rotated
- Some items can be only seen in one view, or partially seen in both views

2.1 | Data loader

We customised the data loader to read the unstructured files, and make labels. To decrease the use of GPU RAM, we resize the input images of each view to 450*600 to reduce the use of computation resources. Data augmentation were initially considered but abandoned eventually, as it required too much GPU RAM at more than 40 GB. In addition, we tested augmentation at a small scale, and it was ineffective, likely because the Dual-View X-ray are overlapped and hidden, also prohibited items can be insignificant (e.g. lighter). Moreover, as the data size is large sufficient, we have the confidence that the model can learn well about each label's pattern (e.g. rotation, monopolies etc.) without augmentation. We build a label vector for each image (pair), with 15 elements, each element represents a prohibited item, and the value is 1 if the item is present, 0 otherwise. All 0 means this is a negative sample (non-prohibited items). The label vector is then used as the target of supervise learning. The whole dataset was then to split into 80% training and 20% validation.

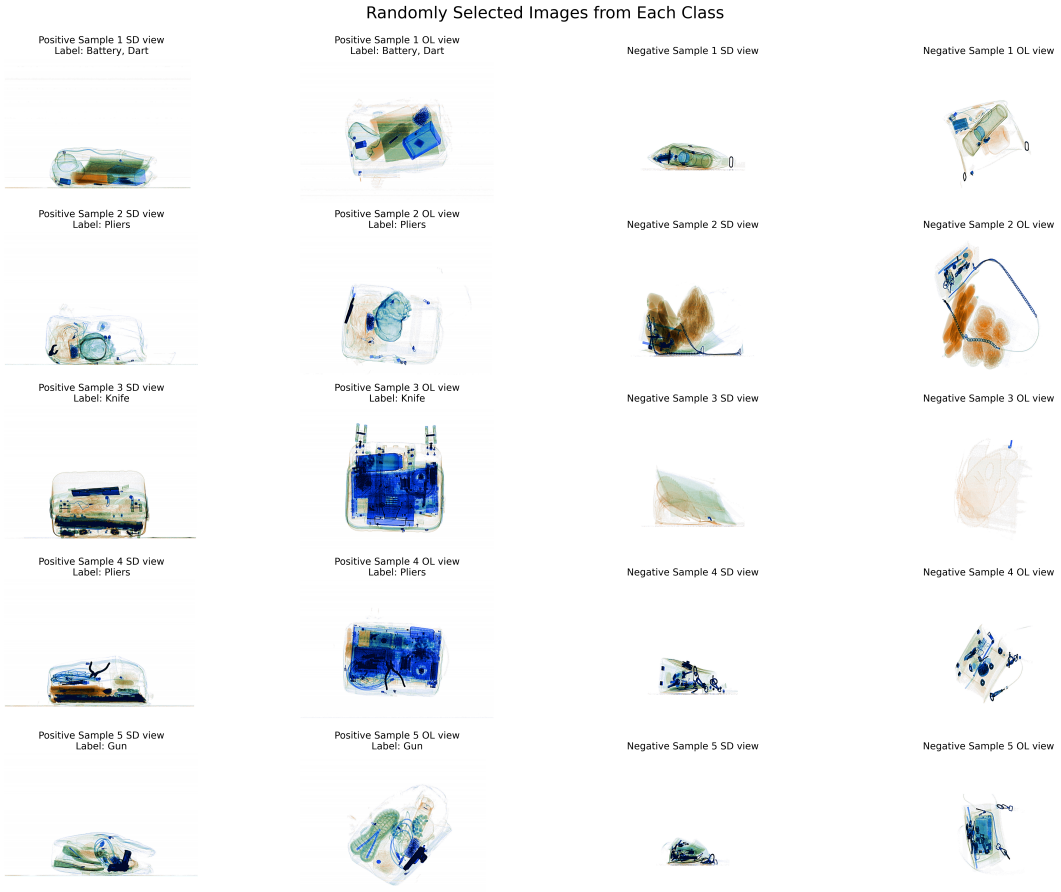

Figure 2.1: Negative and positive samples of the dataset. Each pair contains an orthogonal view and a side view image. Positives may contain multiple prohibited items.(labels)

2.2 | Data errors

Also, during our preprocess of the dataset, wrong labelling has been found, these including missing labels, incorrect labels for single or multiple prohibited items. E.g. in Figure 2.1 negative sample 1, there is highly likely to have a `Battery` while the label is missing. The wrong labelling may affect the model's performance, and make the loss fluctuate.

3 | Methodology

The proposed method is based on a dual-view multi-label classification model, using ResNet50 as the backbone network. The model takes as input a pair of X-ray images (orthogonal and side views) and predicts the presence of each prohibited items in the baggage, ideally, this model can successfully detect prohibited items in the baggage with high accuracy and minimal false negatives. As this is a multi-label classification problem, that is one image (pair) may have more than one label (prohibited items), and those prohibited items are not mutually exclusive. We use the binary cross-entropy loss function to train the model, which is suitable for multi-label classification tasks. The metric used to evaluate the model's performance is the average precision (AP, average of each's class precision (TP/FN+TP)) for each class, which measures the model's ability to correctly identify each prohibited item in the dataset.

3.1 | Network Architecture

The proposed network architecture is based on the ResNet50 backbone, which is modified to accept dual-view input images. The output of the network is a 1×15 tensor, representing the predicted probabilities for each of the 15 classes in the multi-label classification task.

3.1.1 | Input and Splitting

Let $\mathbf{x} \in \mathbb{R}^{B \times 6 \times H \times W}$ denote the input image, where B is the batch size, 6 represents the number of channels (3 for each view), and H and W are the height and width of the image, respectively. The input is split into two views: overlaid view $\mathbf{x}_{ol} \in \mathbb{R}^{B \times 3 \times H \times W}$ and side-by-side view $\mathbf{x}_{sd} \in \mathbb{R}^{B \times 3 \times H \times W}$.

3.1.2 | Feature Extraction

A shared ResNet50 (excluding the last 2 layers) feature extractor $f(\cdot)$ is used to extract features from both views:

$$f(\mathbf{x}_{ol}), f(\mathbf{x}_{sd}) \in \mathbb{R}^{B \times 2048 \times H' \times W'}$$

Where H' and W' are the height and width of the feature tensor, respectively. By concatenating the two views along the batch dimension and passing them through the shared feature extractor, the idea is to learn a shared representation that captures the commonalities between the two views, enabling the model to leverage information from both views for better classification performance.

3.1.3 | Top Layer

This is our layer for regularisation strategies and final feature extraction before the classifier. The shared top layer $g(\cdot)$ further processes the extracted features:

$$g(f(\mathbf{x}_{ol})), g(f(\mathbf{x}_{sd})) \in \mathbb{R}^{B \times 2048}$$

This results in two feature tensors, $\mathbf{u}_{ol}$ and $\mathbf{u}_{sd}$, where the subscripts 'ol' and 'sd' are used for naming convenience and do not imply that these feature tensors only contain information from a single view. The top layer includes 1x1 convolution, batch normalisation, ReLU activation, and adaptive average pooling operations. The batch normalisation layer helps regularisation and stabilise the training process by normalising the input to each layer, and the adaptive average pooling layer reduces the spatial dimensions of the feature tensor to 1x1, allowing the model to capture global information from the features.

3.1.4 | Cosine Similarity

The cosine similarity between the two feature tensors $\mathbf{u}_{ol}, \mathbf{u}_{sd} \in \mathbb{R}^{B \times 2048}$ is computed as:

$$\mathbf{s} = \frac{\mathbf{u}_{ol} \cdot \mathbf{u}_{sd}}{\|\mathbf{u}_{ol}\| \|\mathbf{u}_{sd}\|} \in \mathbb{R}^{B}$$

The cosine similarity measures the similarity between the two feature tensors and helps the model capture the relationship between the views, facilitating information fusion.

3.1.5 | Feature Fusion

The two feature tensors are fused based on the cosine similarity using a weighted sum:

$$\mathbf{v} = (\mathbf{u}_{ol} + \mathbf{u}_{sd}) \odot \mathbf{s} \in \mathbb{R}^{B \times 2048}$$

where $\odot$ denotes element-wise multiplication. Feature fusion allows the model to adaptively adjust the importance of each feature dimension based on the similarity between the views, enabling better integration of information from both views.

3.1.6 | Classifier and Prediction

The fused feature tensor $\mathbf{v}$ is passed through a classifier $h(\cdot)$ for multi-label prediction:

$$\mathbf{p} = h(\mathbf{v}) \in \mathbb{R}^{B \times C}$$

where C is the number of classes (15 in this case). The sigmoid activation function is used to map the predictions to the $[0, 1]$ interval, representing the probability of each class. We also tested adding a dropout layer to help with regularisation before the classifier, but it did not improve the model's performance; therefore, we did not include it in the final model.

The overall network architecture is visualised in the following diagram (Figure 3.1):

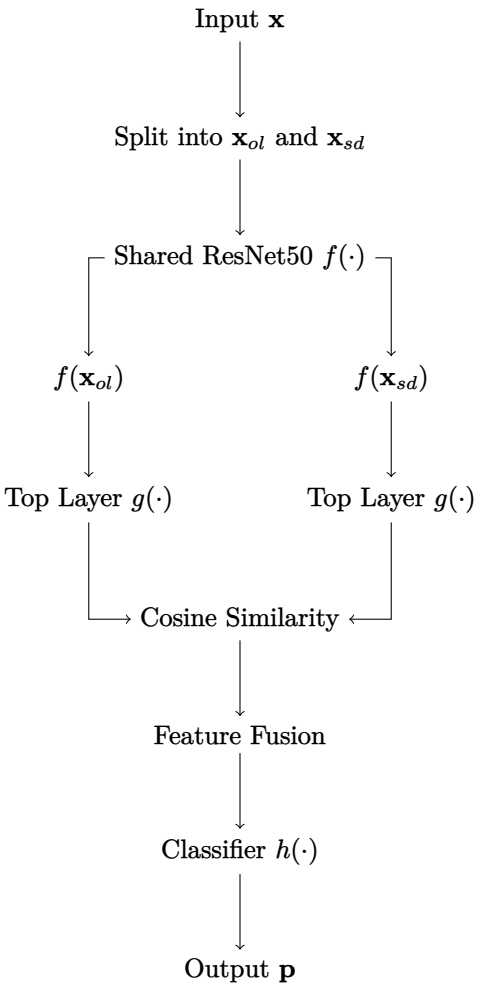

Figure 3.1: Architecture of the DualView ResNet model, showcasing the integration of overlaid and side views through a shared feature extraction and fusion process.

3.2 | Training

The model is trained using the binary cross-entropy loss function, which is suitable for multi-label classification tasks. The loss is computed as the average of the binary cross-entropy loss for each class:

$$\mathcal{L} = -\frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{C} y_{ij} \log(p_{ij}) + (1 - y_{ij}) \log(1 - p_{ij})$$

Where B is the batch size, C is the number of classes, y_{ij} is the ground truth label for class j in sample i, and p_{ij} is the predicted probability for class j in sample i. The model is trained using the Adam optimiser with an initial learning rate of 1.52×10^{-4}, which was determined using the learning rate finder method.

The learning rate is the rate of model's weight to be updated towards minimising the loss function, it is a critical hyperparameter that affects the model's convergence and performance. We used the learning rate finder method to find the optimal learning rate for training the model. The learning rate finder method involves training the model with exponentially increasing learning rates and monitoring the loss. The optimal learning rate is typically chosen as the point where the loss starts to decrease rapidly.

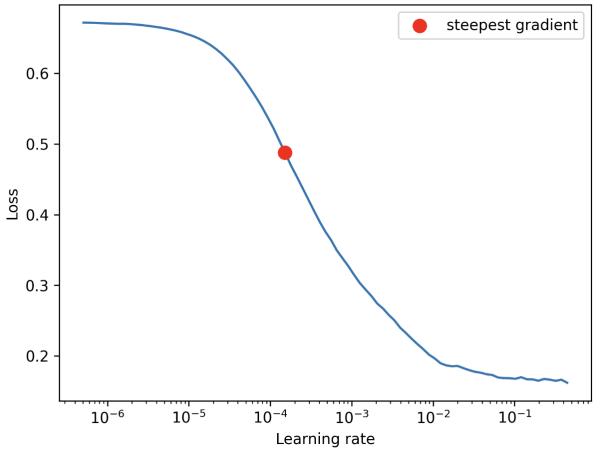

Figure 3.2: Suggested learning rate searched: 1.52E-04

To avoid overfitting, the model is then trained for 30 epochs, which was tested to have the best balance between model generalisation and precision, batch size was chosen to be 32 which nearly maximise the use of our GPU RAM (40 GB), and the `CosineAnnealingLR` scheduler is used to adjust the learning rate during training. The model is evaluated using the average precision (AP) metric for each class, which measures the model's ability to correctly identify each prohibited item in the dataset.

4 | Results

We could observe that the model has generalised well. The loss and AP converges to a stable level (Figure 4.1), with average precision about 91% in validation data set after 20 epochs, and final validation AP 94.4%. Consider there are some wrong labels, this is a very good result. The spike in validation loss and AP are highly possible due to the wrong labelling we mentioned before (subsection 2.2). Compared to original paper's results. Their Vanilla (unmodified) ResNet50 has 67% AP, this is likely due to the Network design difference, their Network is a parallel design where each view has its own feature block, not a shared backbone. This accuracy is expected to further improve with corrected labels and more data trained.

4.1 | Performance Metrics

To dive deeper into the model's performance, we evaluate the model using various performance metrics, including precision, recall, F1 score, micro-averaged metrics, macro-averaged metrics, hamming loss, and subset accuracy on validation set. These metrics provide insights into the model's ability to correctly identify prohibited items in the dataset and its overall performance.

The model's performance on validation set achieved micro-averaged precision, recall, and F1 score of 94.96%, 84.60%, and 89.48%, respectively, indicating that the model performs well in identifying prohibited items in the dataset. The macro-averaged precision, recall, and F1 score are 95.11%, 84.63%, and 89.09%, respectively, showing that the model performs well across all classes. The hamming loss is 0.0045, indicating

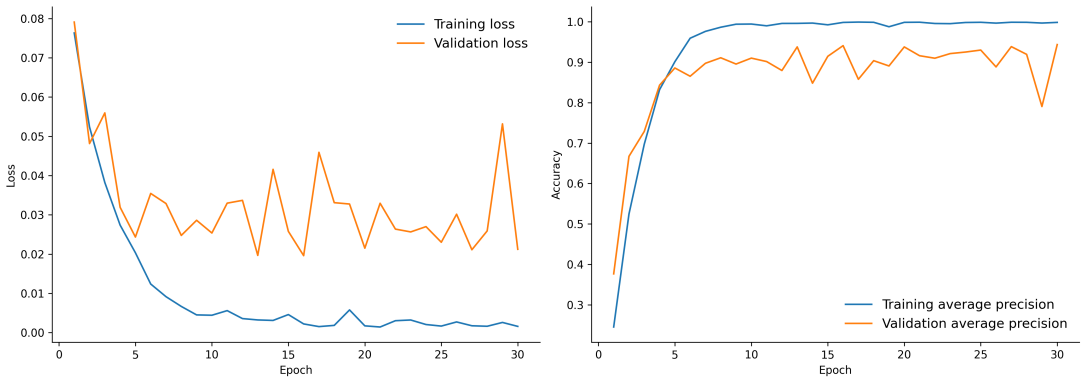

Figure 4.1: Training & Validation Loss and Precision

Prohibited Item	Precision	Recall	F1 Score
Gun	0.960000	0.972973	0.966443
Knife	0.870968	0.692308	0.771429
Wrench	1.000000	0.890244	0.941935
Pliers	0.926471	0.954545	0.940299
Scissors	0.959184	0.652778	0.776860
Lighter	0.934426	0.802817	0.863636
Battery	0.987013	0.883721	0.932515
Bat	1.000000	1.000000	1.000000
Razor blade	0.978261	0.642857	0.775862
Saw blade	0.900000	0.830769	0.864000
Fireworks	1.000000	0.925926	0.961538
Hammer	0.987013	0.987013	0.987013
Screwdriver	1.000000	0.691176	0.817391
Dart	0.782051	0.802632	0.792208
Pressure vessel	0.981818	0.964286	0.972973

Table 4.1: Multi-Label Classification Report on Validation Set

that the model makes few errors in predicting the presence of prohibited items. The subset accuracy is 94.25%, showing that the model accurately predicts the labels for most samples. Subset accuracy measures the proportion of samples where all the predicted labels exactly match the true labels. It is a strict metric that requires the predicted label set to be an exact match of the true label set for each sample. Subset accuracy is calculated as: (Number of samples with exactly matching labels) / (Total number of samples)

Where the micro-averaged metrics are calculated by considering all samples together, and the macro-averaged metrics are calculated by averaging the metrics for each class. The hamming loss measures the proportion of incorrectly predicted labels.

4.2 | Confusion Matrix

We also provide a multi-Label confusion matrix (Figure 4.2) to visualise the model's performance in identifying prohibited items in the dataset. The confusion matrix shows the number of true positives, false positives, true negatives, and false negatives for each class, providing insights into the model's performance for each prohibited item.

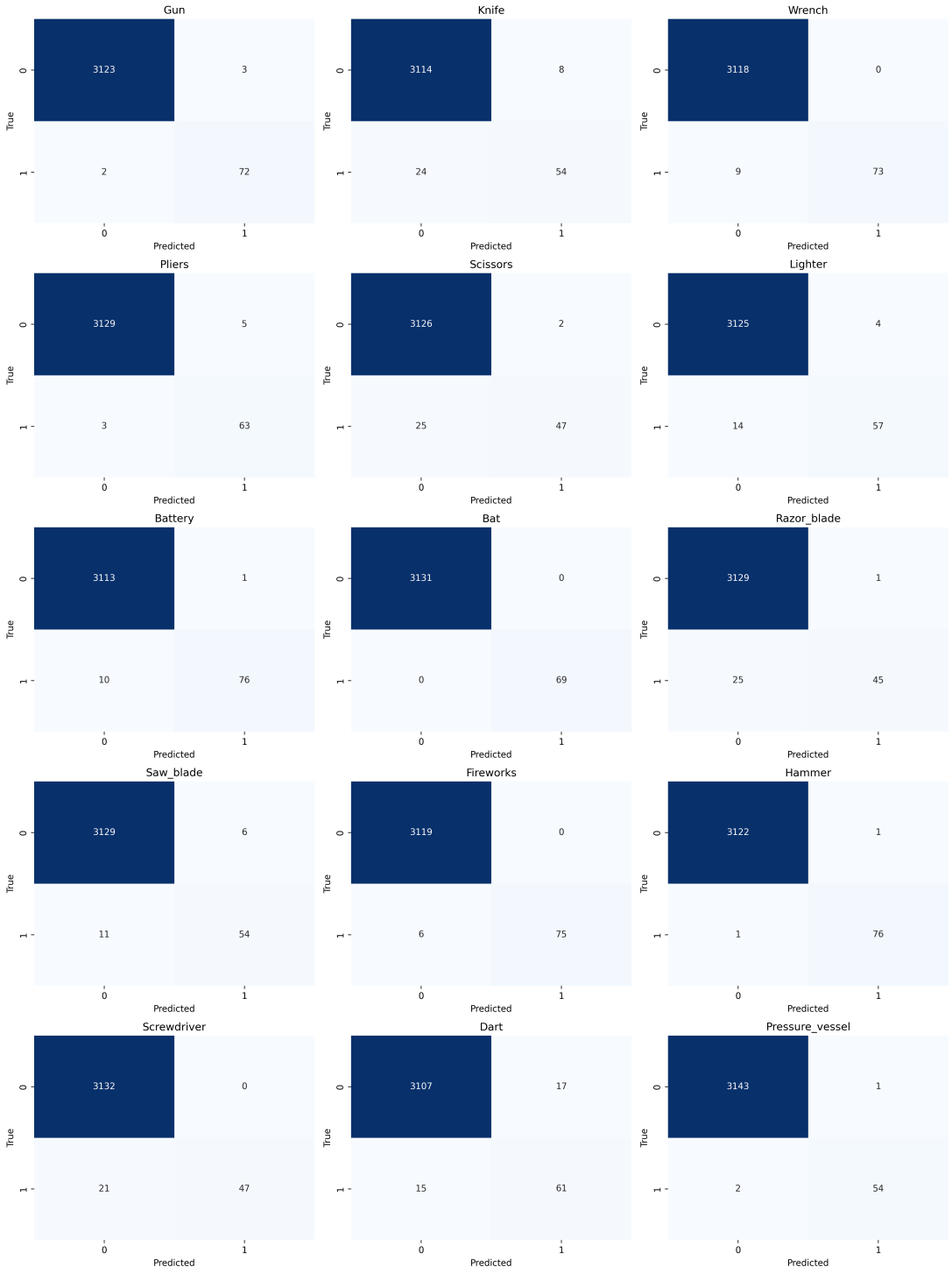

Figure 4.2: Multi-Label confusion matrix

4.3 | Error Analysis

To further look into the model's performance in more detail, we analyse the errors made by the model in predicting the presence of prohibited items in the dataset. We identify the most common types of errors made by the largest loss in the validation set.

From the Figure 4.3, we can see that the model mainly struggles with detecting small and partially visible prohibited items, such as lighter, scissors, and often confused between similar items like knife and saw blade.

Figure 4.3: Error Analysis, this image shows the largest 32 loss samples in validation set.

5 | Conclusions

The automated baggage screening system utilises a deep learning model based on a ResNet50 architecture, trained with dual views, which has shown promising results by accurately identifying prohibited items within baggage. This system is capable of identifying multiple prohibited items in one sample and handling multiple overlapping items that are rotated to different angles, which is a common situation in real-world applications. Ideally, this model could employ in security systems to improve the efficiency and accuracy of baggage screening processes, and reduce human error in detecting prohibited items.

The key features of this model is that it demonstrates a high subset accuracy of 94.3% and a high average precision (AP) of 94.4%. It efficiently processes dual views and shows high noise resistance despite incorrect labelling in the data. However, the performance of the model is somewhat unstable due to incorrect labelling in the training data. Memory constraints prevented the use of data augmentation before training, suggesting that a larger dataset could further improve the model's performance.

6 | References

Ma, B., Jia, T., Li, M., Wu, S., Wang, H., & Chen, D. (2024). Toward dual-view x-ray baggage inspection: A large-scale benchmark and adaptive hierarchical cross refinement for prohibited item discovery. *IEEE Transactions on Information Forensics and Security*, *19*, 3866-3878. doi: 10.1109/TIFS.2024.3372797