Irregular luggage classification experiments using data from Tromsø airport

Andreas Dyrøy Jansson
Department of Computer Science and Computational Engineering
UiT The Arctic University of Norway
Narvik, Norway
andreas.d.jansson@uit.no

Bernt Arild Bremdal
Department of Computer Science and Computational Engineering
UiT The Arctic University of Norway
Narvik, Norway
bernt.a.bremdal@uit.no

Espen Remman
Chronos AS
Oslo, Norway
espen.remman@chronos.ai

***Abstract*—This paper examines the current state-of-the-art in object tracking and detection in the context of baggage handling systems using a variety of technologies, with special interest in image classification. Data collection at Tromsø Airport using simple and inexpensive equipment in order to capture images of different types of luggage will be described. Furthermore, a selection of three relevant techniques for image classification will be investigated, and experiments will be conducted to detect irregular and potentially problematic bags in the collected dataset. Finally, recommendations for use cases and an interface to an intelligent baggage handling system will be proposed.**

***Keywords*— baggage handling, convolutional neural network, data collection and preprocessing, image classification, random forest, support vector machine**

I. Introduction

Perhaps the most important task of an airport is to ensure that passenger luggage is safely and efficiently transported to their designated destination. Failure to deliver bags on time may lead to dissatisfied passengers, extra work for airport personnel, costly delays, and missed flights. Delays may even have an environmental impact in the form of increased fuel consumption of both airplanes and ground vehicles during bottlenecks and queues [1]. According to SITA, more than half of all bag delays were caused by either transfer mishandling or failure to load the bag [2]. In 2018, IATA introduced Resolution 753 as an effort to reduce the number of lost and mishandled bags, by mandating members to keep an accurate inventory of all items entering and leaving the baggage handling system (BHS) [3]. Airports may comply by using manual labor, or by increased use of sensors, smart systems, and automation. Tracking and reporting luggage status manually is time consuming and tiring work, especially in larger airports. On the other hand, an intelligent autonomous BHS would be able to monitor and track luggage in multiple locations, detect problems, provide useful insight to personnel, and may even provide added value to passengers in the form of real-time tracking and status updates. Such a system is envisioned to consist of multiple intelligent entities, some of which may include robot arms for picking up and stowing luggage, cameras, autonomous baggage carts, and a set of software agents monitoring status and allocating resources based on the current load. A BHS has many similarities to a shipping facility, and one may draw inspiration from companies like Norway's Komplett [4] or the global giant Amazon [5], where the package dispatch, shipping and tracking is managed by autonomous systems. However, unlike standardized parcels used in shipping, personal luggage may be of a variety of shapes, sizes, and textures. As a result, an intelligent BHS requires accurate information about incoming items, which in turn creates the need for relevant data sets. Visual recognition of luggage is expected to be an appropriate method, given the amount of research that has been performed on object detection and classification in recent years. As such, this paper will examine image data collection, preprocessing, and labelling, before investigating and testing a selection of classification models. Furthermore, the proof-of-concept should be cost effective and unintrusive, making adoption less of a hurdle. This means that both the data collection hardware/sensors, required infrastructure, and the model itself should be light-weight and inexpensive, and be made of readily available components.

II. Related work and contributions

Efforts have been made with respect to both tracking and classification of luggage using various technologies. Examples include the use of smart tags based on radio frequency identification (RFID) to improve bag tracking precision [6, 7], and owner verification during baggage claim [8]. In short, it has been shown that RFID is a reliable method of tracking luggage. However, this requires additional hardware and supporting infrastructure, which in turn requires effort from both passengers and airport personnel to enable successful adaptation. On the other hand, security systems based on video surveillance are already in widespread use in airports today and extending such a system with the necessary components for luggage tracking and classification is expected to require minimal infrastructural upgrades.

Indeed, examples of image recognition in the context of classification and tracking of luggage are shown in [9-16]. Convolutional neural networks (CNNs) were used in [9, 10] to identify and classify abandoned items. In [11], a multi-camera setup with transfer learning was used to track passengers and their luggage at security checkpoints in an airport. Perhaps the most well-documented application of object recognition and image processing in relation to airports and luggage is the detection of illegal items in x-ray images from baggage scanners [12-15]. Examples include automatic detection of suspicious objects using both deep and shallow networks, stacked autoencoders, and random forest [12]. In this paper, the authors investigated and compared a selection of algorithms for detecting firearm parts. In [13], multiple views of the x-ray machine were combined to detect illegal objects from various angles. Furthermore, in [14], the authors demonstrated how free and open-source software could be used for classification of x-ray scans in an airport, using TensorFlow with CNNs and transfer learning. In [15], detection of small metallic items in high-resolution, cluttered images of shipping containers was performed using CNNs.

Finally, in [16], the authors investigated the use of static filters for image classification of abandoned objects in an airport using security camera feeds. The authors were able to devise a set of domain-specific filters, as there were only so many objects and features one can expect to appear at an airport. They showed that a relatively simple set of features were sufficient in order to classify an object with a high success rate using a certain combination of filters, and how many times they appeared in each object.

In addition to classification of luggage, how to route and model bag flow in a BHS have also been documented various papers. For instance, how to deal with prioritized items, bag jams and coordination of different bag origins was investigated in [17, 18]. A more detailed, event-driven simulation based on the concept of Petri Nets was presented in [19]. The main area of interest in this paper was to identify and analyze the percentage of bags not reaching their intended destination in time, which the authors refer to as “failed bags”. One cause of failed bags is the fact that paper tags are prone to failure and may become mangled beyond recognition, leading to manual intervention and thus risk being late for their destination.

Based on these findings, this paper will explore the following topics:

- How can cameras be placed in suitable locations along a BHS to capture and create a dataset for baggage type classification?
- How can the collected dataset be utilized for counting and detection of potentially problematic luggage?
- Is it possible to achieve precise classification of irregular luggage items without using deep models?

As such, the contributions of this paper are as follows: first, acquisition of relevant data will be performed, and findings will be presented. Next, a selection of relevant image processing techniques will be discussed and compared, followed by a set of classification experiments on the collected images of luggage. A convolutional neural network, a support vector machine, and a random forest classifier will be trained and evaluated. Finally, the results will be discussed, and suggestions for future work and potential applications and benefits in the context of an intelligent BHS will be addressed.

III. Approach

An intelligent BHS as proposed in the introduction may essentially be controlled using either a centralized, top-down structure, or a distributed, edge-based architecture consisting of smart and/or autonomous agents representing various components. Centralized control may be easier to design and implement in the short run, but may lack scaling flexibility, in addition to communication overhead. The master controller in a centralized system would have to be powerful enough to process all inputs from all sensors, make decisions based on the data, and communicate instructions back to each part of the BHS. Transferring and processing complex sensor data in the form of images requires both robust and low-latency communication infrastructure, and powerful computational hardware. Furthermore, this architecture is more susceptible to failure, as it relies on a single point of failure in the form of a central data processing and decision-making device.

On the other hand, in a distributed, edge-based BHS decision system, sensors and actuators can be made “smart” and perform data collection, processing, and decision making locally with low latency. Communication between different parts can be performed ad-hoc, with simplified, standardized and light-weight protocols. A system modelled as a collection of autonomous agents is also expected to be more scalable and robust. Even if one agent, or a group of agents should fail, the whole BHS will not need to be taken offline for maintenance. High level monitoring and control overrides may still be issued by a simpler, centralized controller, but this will play a smaller, non-critical role in a distributed system. As such, this paper will take special interest in edge computing, local data collection and processing, and light-weight classification models.

Today, airports are well equipped with surveillance equipment and related infrastructure. Therefore, an approach leveraging existing hardware is expected to be more cost effective, while simultaneously being less obtrusive. However, all airports may not have surveillance cameras targeting the BHS conveyor belts. As such, the proposed solution involves placing dedicated cameras in strategic locations along a BHS conveyor belt, configured to capture images of each item from various angles. As mentioned previously, an intelligent BHS needs detailed information about incoming items, and the first matter of business is to determine whether the bag is of regular or irregular shape. This assumes that most bags are square, hard, and of a certain size, implying that each part in an intelligent BHS should be optimized for this type of item. On the other hand, it is assumed that a smaller number of bags do not conform to these specifications, in particular backpacks, duffel bags, and ski bags, requiring special care when handled by a non-human actor. The ability to automatically identify and classify irregular baggage for optimized sorting and stacking may even be used to prioritize luggage based on “regularity”, in order to have simpler items that are less likely to cause problems and jams get priority in the BHS, while bags that are identified as “irregular” may have to be sent last to avoid tangles and jams. An overview of the proposed approach is shown in Fig. 1, and each step will be discussed on detail in the following sections.

A. Data collection

As mentioned in the introduction, one of the main areas of interest of this paper is the use of inexpensive hardware and readily available software to perform both data collection and classification. As such, a Raspberry Pi 4B along with off-the-shelf web cameras and external drives were used to collect the image data. The Raspberry Pi was able to handle two

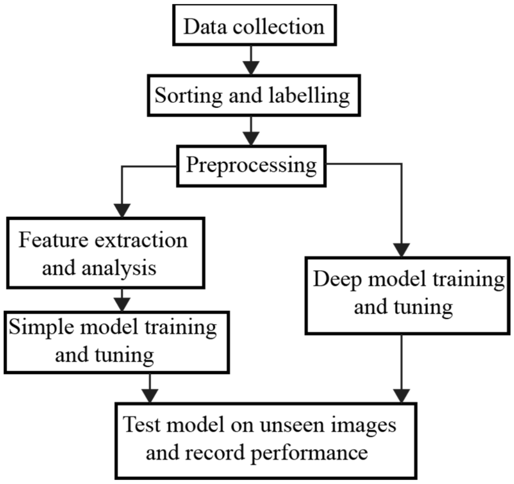

Fig. 1. Block diagam of proposed combinations including all steps

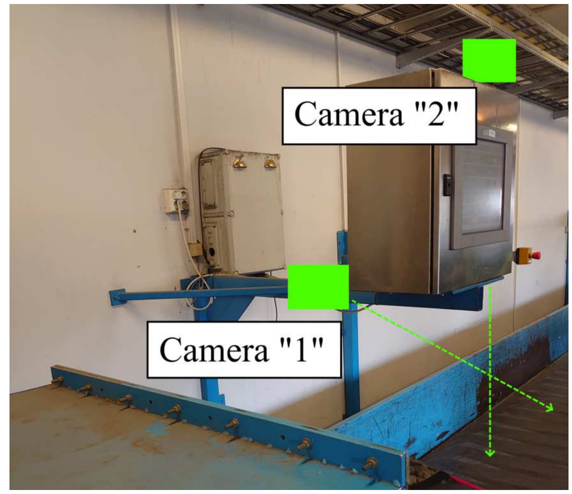

Fig. 2. Placement of data collection rig and cameras at Tromsø airport

simultaneous camera feeds, which were monitored in separate threads using OpenCV [20]. A python script was developed and configured to capture images of resolution 320x240 whenever the mean square error between frames was larger than a manually fine-tuned threshold. The idea was that an empty conveyor would have small changes between frames, as opposed to more significant changes whenever a bag entered the frame and moved past the field of view of the camera. To set up and collect the image data, Tromsø Airport was chosen due to its location, in addition to ease of access to bag conveyors due to undergoing renovation of its BHS.

During renovation, a temporary bag arrivals area was set up, and it was decided to mount the Raspberry Pi at this location. Bags from arriving planes were manually loaded from trolleys onto the belt seen in Fig. 2. The Raspberry Pi and its peripherals, along with one camera was secured to the above cable ladder. The camera provided a top-down view of the conveyor belt, and will be referred to as camera 2 going forward. The other camera, referred to in this paper as camera 1, was mounted on the horizontal support beam seen in the center of Figure 1 for a front perspective view of the same area. The data collection period lasted for three weeks, from February 1st to February 23rd.

An important point to consider when performing any activity on airport premises is security. As such, it was agreed upon that the raspberry pi should not be connected to a network but should only be accessed locally through the hotspot and SSH. If improperly configured, IoT devices may pose severe threats to a network's security and integrity. This was also one of the reasons storing the captured image data locally was chosen in favor of network storage. Furthermore, the GDPR imposes restrictions on the collection and use of any data that may contain personally identifiable information. Most importantly, stowers or other personnel may inadvertently be caught on camera when loading bags onto the conveyor belt. Secondly, the bags themselves could be used to identify their owner directly or indirectly, as it is not uncommon to adorn luggage with identifiable markings or similar embellishments. In addition, name tags containing personal information may be visible and readable. However, if images were captured in a lower resolution, this would be less likely to pose a problem.

B. Dataset preparation and labelling

During the data collection period, a total of 132,768 images were captured from both cameras combined. First, images only containing an empty conveyor were removed from the training sets. Next, a manual review was conducted to identify and remove any images containing identifiable stowers. For the labelling, bags were deemed irregular based on shape, size, loose straps, and texture according to the previously stated assumptions.

Finally, images were manually labelled as either "regular", or "irregular" based on the presence of one or more such irregular bag somewhere in the frame. For camera 1, this resulted in a dataset consisting of 28,348 images of regular items only, and 17,834 images containing irregular items. For camera 2, the dataset contained 16,141 images of regular items, and 9,062 images containing irregular items. Fig. 3 shows examples of images containing both irregular and regular items (a), a single irregular bag (b), a single regular item (c), and multiple regular items. Finally, images were cropped to 240 by 240, resulting in more of the region of interest filling the frame. Images taken with camera 2 had empty regions in the lower section of the frame, which cropping did not remove.

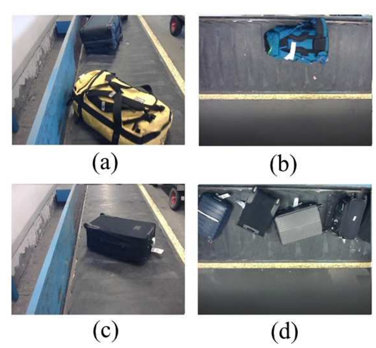

Fig. 3. Various types of images of bags, taken with camera 1 (a, c) and camera 2 (b, d)

C. Image classification experiments, validation and testing

Image processing for object detection may be performed using a variety of computer vision techniques. As previously mentioned, the main area of interest is on detecting irregular pieces of luggage in each image frame and evaluating the performance of computationally expensive versus lighter techniques. For the experiments described in this paper, the following test suite was developed: First, 10% of images from each class, for each camera angle were randomly selected and put aside for testing each model after training. Next, each model was trained on the remaining 90% of images using appropriate methods, described in detail in the next sections. Finally, each model was used to classify the previously unseen 10% of images from their respective angle, and any misclassified images were recorded for later analysis. Training accuracy and prediction results of each model will be discussed in the results section.

1) Convolutional Neural Network A basic CNN classifier was implemented using TensorFlow [21]. The training data was loaded and split 80/20 for training and loss validation using built-in utilities from Keras. For the first experiment, a simple network consisting of three relu-activated 2D convolutional layers, three max pool-layers and two dense layers was implemented. For the CNN experiments, network depths ranging from 3 to 6 convolutional layers were tested, and their respective performance recorded. Each variant of the CNN was trained on images from camera 1 and 2 in separate operations.

2) Manual feature extraction Unlike a CNN, SVM and Random Forest models do not consider the spatial relationship of features in an image [22, 23]. According to [24], feature extraction may be beneficial for classification accuracy in these models. As a result, a preprocessing step was introduced in order to transform the images into a format more suitable for these methods. As seen in [16], extracting features based on domain knowledge showed promise, and a similar method was used for the next experiments. First, images were converted to grayscale. Next, basic features were extracted using the Oriented FAST and Rotated BRIEF (ORB) detector, Harris corner detector, the Shi-Tomasi corner detector (dubbed "GoodFeatures" internally in OpenCV), and Hough line detector. The ORB detector is a keypoint detector and matcher similar to SURF and SIFT [25]. The Harris corner detector detects corners, as defined as a region in an image where the intensity varies in all directions [26]. The Shi-Tomasi corner detector is an alternative corner detector, and may be said to be more restrictive, as it finds the most prominent corners in an image [27]. Preliminary tests showed that images containing irregular bags were more visually complex, and thus contained more features than their plainer regular counterparts. For the training images, feature extraction and counting were performed once in a separate step before training the models, and the resulting values were stored offline in a tabular format. As with the CNN, images from camera 1 and 2 were handled seperately, resulting in two distinct data sets which were later used for fitting the models. On the other hand, test image features used for classification were extracted during runtime, reminiscent of online operation. This process is represented visually in Fig. 4. For irregular items, blob features tended to be more clustered together on the item itself as seen highlighted in green in Fig. 5 (a, b). On the other hand, blobs were usually more spaced out for regular items as seen in (d, e). Furthermore, for the regular bags, a large portion of the detected features were actually on the conveyor belt, as seen in Fig. 5 (d-f). As a result, background subtraction was used to further to highlight the item of interest in the frame. Furthermore, Fig. 5 (c, f) shows features detected using the Harris corner detector for irregular and regular items, marked with red. More specifically, the image (c) containing an irregular item had a total of 43 Harris corners, as opposed to 25 for the regular items only (f). It is also possible to see the features are more clustered together on the irregular item itself (Fig. 5 (c)), while they are more scattered in Fig. 5 (f), similar to the blob features. Next, a preliminary investigation of the separability of the data points was conducted by plotting two and two features in 2D and trying to identify obvious clusters. This will be discussed in more detail in the results section.

3) Support vector machine For the SVM classification, the Support Vector Classifier from sklearn [28] was used. The model was fitted on the extracted features using the default parameters, as described in the documentation [29]. To test the model, features were extracted and counted, and then fed to the model for classification. The number of misclassified bags of each class was recorded, and will be presented in the results

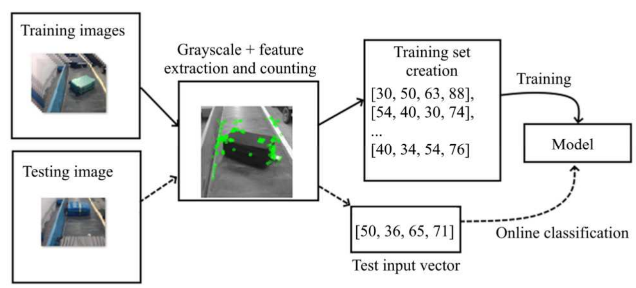

Fig. 4. Image feature extraction and model training process

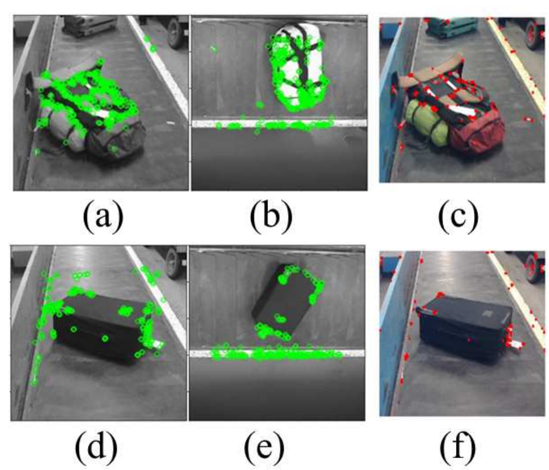

Fig. 5. Examples of detected ORB features (green) and Harris corners (red)

4) Random forest For the last set of experiments, the Random Forest classifier from sklearn [28] was used for the classification. An initial experiment was conducted in order to find the optimum number of trees in the forest, testing forests consisting of 50-200 trees. Performance of each forest was recorded, and the resulting optimum forest size was used for the concluding experiment. In addition, confusion matrices for both SVM and Random Forest classification were produced for better understanding of the outputs.

5) Final analysis In the interest of understanding the reasoning behind each model's decision, images that were misclassified by all three models were put aside for manual review after each experiment. Examples of such images will be presented in the results section.

IV. Results

A. Image classification using CNN

For the first experiment, no data augmentation or tuning was performed beyond creating a basic network structure as described in the previous section, and the training and validation accuracy and loss for epochs 0 – 5 are plotted in Fig. 6. Classification accuracy on the training set was close to 100%, while the test set was a mere 78%. Furthermore, the loss of the network on the training set was steadily decreasing for each epoch, while the loss on the validation set increased significantly in the last few epochs.

After the depth was increased to 4 convolutional layers, validation accuracy improved slightly. Increasing the depth to 5 layers showed even more promise, bringing validation accuracy closer to the training accuracy. Finally, adding a 6th layer did not contribute to significantly improve validation accuracy. Based on these findings, a 4-layer network trained for 2 epochs was used for the final classification test. Using this architecture, the misclassification rate was 15.3% and 23,7% for camera 1 images of regular and irregular bags, respectively. Similarly, for images taken with camera 2, the misclassification rate was 8.6% and 38.8%, respectively.

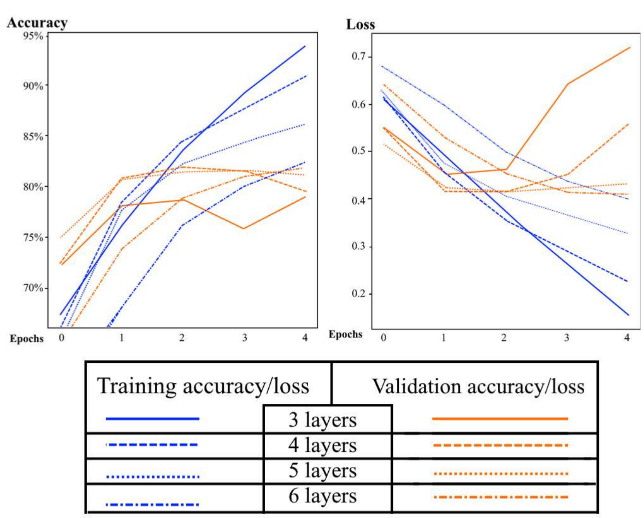

Fig. 6. CNN classification accuracy and loss for layers 3-6

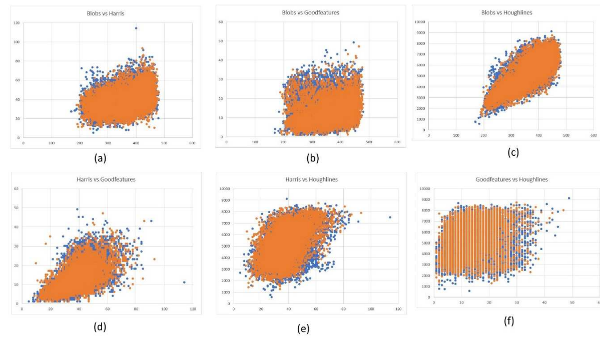

Fig. 7. Extracted features of each class (orange and blue) represented in 2D

B. Extracted features analysis

Next, an investigation of the separability of the data points was conducted by plotting two and two features in 2D and trying to identify obvious clusters. At first glance, there was no clear linear separation or clusters in the plots in Fig. 7 (a-f), as points from both classes (orange and blue) have large areas of overlap.

C. Image classification using SVM

Using the SVM-classifier on the extracted features resulted in a 11.8% and 72.9% misclassification rate for camera 1. The associated confusion matrix presented in Fig. 8 shows the full picture, and this will be discussed in more detail in the next section. For camera 2 images, a similar misclassification rate of 8.0% and 73.0% was observed for regular and irregular items, respectively.

D. Image classification using Random Forest

Random forests with 50-200 trees were tested, and the performance of each forest was recorded. Based on the resulting best accuracy, a forest with 178 trees was used for the testing experiment. This resulted in 24.7% of images of regular items and 59.2% of images containing one or more irregular bag being misclassified. Finally, for camera 2 images, a misclassification rate of 19.1% and 57.3% was recorded for regular and irregular items, respectively. As with SVM, a confusion matrix was produced, which is shown in Fig. 9 and will be discussed later.

SVM			
Total image count 4234		Predicted class	
		764 Irregular	3470 Regular
Actual class	1739 Irregular	470 Irregular classified as irregular	1269 Irregular classified as regular
	2495 Regular	294 Regular classified as irregular	2201 Regular classified as regular

Fig. 8. Confusion matrix for camera 1 images classified with SVM

Random Forest			
Total image count 4234		Predicted class	
		1326 Irregular	2908 Regular
Actual class	1739 Irregular	710 Irregular classified as irregular	1029 Irregular classified as regular
	2495 Regular	616 Regular classified as irregular	1879 Regular classified as regular

Fig. 9. Confusion matrix for camera 1 images classified with Random forest

E. Analysis of misclassified bags

In the end, the same 26 images of regular items and 380 images containing irregular items were misclassified by all three models. Of the images containing one or more irregular items, 112 images were of irregular items only, while the remaining 268 featured regular items. A selection of offending images is presented in Fig. 10.

V. Discussion of results

Implementing a rudimentary luggage classifier based on CNN using Keras and Tensorflow was fast and simple. As the first iteration of the network only used three convolutional layers, accuracy was low. This may be due to the image size used, and the images themselves being of a complex and noisy nature. As seen in Fig. 6, introducing more layers did improve accuracy, which may also be due to the image complexity. However, introducing more layers may cause the network to overfit, and since adding more than 4 layers did not improve the accuracy much in this case, a network with 4 layers became the best compromise between accuracy and computational expense. However, one can argue that a model is only ever as good as the data it is trained on. As no image processing, enhancing or other techniques besides cropping were used on the training images, there may be even more gain to be had in this regard. As shown, the images used for this paper were collected and used directly in the models, relying only on appropriate labelling. As a result, more care could be taken to clean and preprocess the data for better classification accuracy. This is even more true for the other models used in this paper.

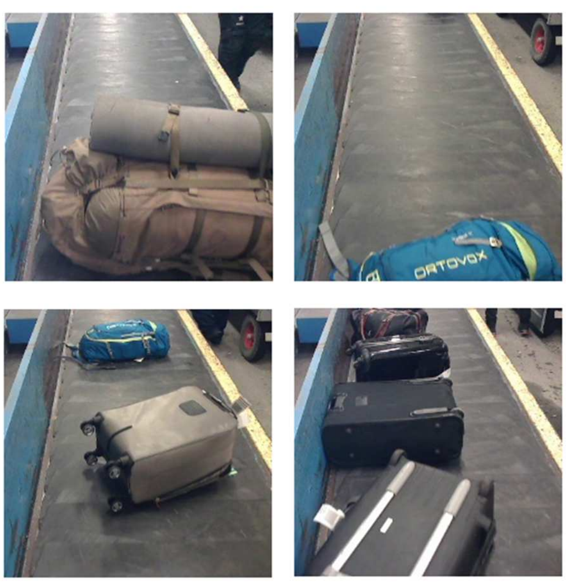

Fig. 10. A selection of camera 1 images labelled as "irregular" misclassified by all models

Support Vector Machines also proved to be a viable option for classifying images of luggage, albeit with extra preprocessing and feature extraction steps. Advantages of SVM over CNN include transparency, which may be desirable when predictions are to be used in a decision system, like the one proposed here. In addition, the models are simpler and less computationally expensive to train and run, which may be desirable in an edge setting in a decentralized BHS. Furthermore, in this particular case, the SVM was able to predict the class of regular images with higher accuracy than the CNN, which is rather interesting. However, this may be due to dataset imbalance, as any model would be more likely to predict the correct label by chance due to the larger sample size. As with CNNs, care should be taken to preprocess and understand the data properly, and not rely solely on tweaking the model itself for performance. It was found that the SVM was able to correctly classify 88% of images of regular bags, but only 27% of the images containing one or more irregular bag. In other words, the model performed worse than random guessing. One part of the explanation may be that the dataset was imbalanced in favor of regular items. However, Fig. 8-9 show a 4.5 to 1 imbalance, which is significantly more than the 1.4 to 1 ratio of the training data. Statistically, one may assume that the imbalances would be more similar in magnitude. This discrepancy may be caused by the way images were labelled, as any image containing only a small visible part of an irregular item, it was labelled as such. In other words, an image containing an obstructed view of an irregular item behind two or three regular bags would be labelled as "irregular". This hypothesis was further strengthened when looking at the actual misclassified images, like the ones shown in Fig. 10. As mentioned, for the irregular images misclassified by all models, 70% also prominently featured regular items that might throw off the predictions.

Similarly, random Forest was fast and computationally cheap to train, and produced results comparable to the SVM. In this case, the random forest model was able to more accurately predict the class of images containing irregular items, but worse on the regular items. For the images containing irregular items, the model was even worse than random guessing. Depending on the depth and complexity of each tree, decisions may be difficult to validate. However, as random forest is a transparent model, it is still easier to justify than the decision made by a CNN.

Although not terrible, the classification accuracy of the models still leave room for improvement. As seen, common for all models examined was that they struggled to correctly classify irregular items, with the lowest misclassification rate of 23,7% being achieved by the 4-layer CNN. However, even with minimal optimization and only basic preprocessing and tuning, preliminary results did show some promise, and further investigations should be made into at least one of the models presented.

A. *Further work and recommendations*

As mentioned, greater care should be taken in order to extract appropriate features and increase classification performance. For a start, the dataset could be further processed and distilled into a more concentrated state, making it easier for a model to distinguish between the classes. One possible solution may be to simply remove any image not prominently featuring an irregular bag. However, removing too many such images would further imbalance the dataset, which may in turn be detrimental for generalization. Another recommendation is to investigate the use of bounding box labelling, which is expected to help single out irregular bags in an otherwise crowded image. Different filters beyond the ones presented in this paper should be investigated, with emphasis on identifying features exclusive to images of irregular bags.

In addition, in order to properly label the images collected at Tromsø, interviews with stowers should be conducted. In the experiments presented in this work, a set of assumptions were made that may not hold true according to domain experts. Even if classification performance in this particular instance left much to be desired, investigations should be made to see if it would be possible to perform a portion of the labelling work based on the models and filters presented in this work using transfer learning.

Even if an item was classified as "regular", it may become dislodged, or even fall off the conveyor, becoming lost or creating other problems for autonomous systems down the line. As such, additional classes based on the state of a bag could also be introduced for further insight and intelligence. Again, domain experts should be interviewed to determine the most appropriate states. Instead of a binary classification, luggage items could be given a score based on how likely they were to cause a jam, or other problems. Items with a low probability would then get priority in the BHS. This would however require a mechanism for assigning and keeping track of the priority of individual bags, requiring additional, potentially expensive and complex infrastructure which is beyond the scope of this work.

For more accurate classifications, there is also the possibility of combining the image data with other measurements, including the weight of the item as detected by the conveyors, or other external data sources. For example, if there is a vacation coming up, a lot of people may be traveling with sports equipment, and so on. In this particular instance, a number of the collected images featured ski bags and large backpacks, which is a result of the time of year data was collected, and the airport location itself, as Tromsø is a popular winter destination in northern Norway. As such, the rig could be configured to tag images with metadata during data collection, examples include flight information, bag origin, time of day, season, and destination. If the image resolution was high enough, paper baggage tags could also be captured and decoded by the system for additional metadata. This in turn may be useful for future endeavors.

B. *Interface to an intelligent BHS*

On the departure side, passengers check in and drop off their bags at designated locations as normal. Cameras could then register when a new bag enters the system, which allows consistent tracking going forward. As a result, in the case of a tangle, jam, or other unforeseen event causing bags to become lost, personnel are better equipped to track down the offending item. On the arrival side, the process would be performed in reverse. Bags would be unloaded from the plane onto autonomous trolleys and transported back to the terminal where they are picked up and placed on conveyors by robot arms. As discussed in the introduction, a good portion of lost bags are a result of errors during transfer. Cameras placed alongside the conveyor in an intelligent BHS would make it easier to confirm whether a bag arrived at its transfer destination at all, if it was transferred successfully to the correct plane, or if it became lost somewhere along the way. An accompanying service may also be offered to passengers where they could report a bag as lost using a picture of the bag, potentially helping personnel identify the bag faster.

Acknowledgment

The authors wish to thank the stowers at Tromsø airport for their time and allowing the data collection rig to be set up in the temporary baggage arrivals area.

References

[1] F. Zhou, G. Jiang, Z. Lu, and Q. Wang, "Evaluation and analysis of the impact of airport delays," *Scientific Programming*, vol. 2022. Article ID 7102267, 2022, https://doi.org/10.1155/2022/7102267

[2] SITA. "Baggage IT Insights." SITA.aero. https://www.sita.aero/resources/surveys-reports/baggage-it-insights-2022/ (accessed Nov. 20, 2022).

[3] IATA. "Baggage Tracking, IATA Resolution 753/A4A Resolution 30.53 Implementation Guide"

[4] E. H. Urke. "Unik video: Bli med inn i Kompletts splitter nye robotlager." Tu.no. https://www.tu.no/artikler/unik-video-bli-med-inn-i-kompletts-splitter-nye-robotlager/223905 (accessed Mar. 3, 2023)

[5] K. Jackson. "Amazon has new robots joining its warehouse workforce." CNET.com. https://www.cnet.com/tech/amazon-robots-will-join-its-warehouse-workforce (accessed Mar. 3, 2023)

[6] F. Wang, P. Si, E. Sun and Y. Su, "BEI-TAB: Enabling Secure and Distributed Airport Baggage Tracking with Hybrid Blockchain-Edge System," *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, Tianjin, China, 2021, pp. 1221-1225, doi: 10.1109/ICCT52962.2021.9658084.

[7] S. Mul, A. Philip, M. Correia and L. Gadhikar, "Baggage Tracking Using RFID and Blockchain Technology," *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, NaviMumbai, India, 2021, pp. 1-5, doi: 10.1109/ICNTE51185.2021.9487770

[8] A. H. Salman, T. Adiono, I. Abdurrahman, Y. Aditya and Z. Chandra, "Aircraft Passenger Baggage Handling System with RFID Technology," *2021 International Symposium on Electronics and Smart Devices (ISESD)*, Bandung, Indonesia, 2021, pp. 1-5, doi: 10.1109/ISESD53023.2021.9501689

[9] S. Smeureanu and R. T. Ionescu, "Real-Time Deep Learning Method for Abandoned Luggage Detection in Video," *2018 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, 2018, pp. 1775-1779, doi: 10.23919/EUSIPCO.2018.8553156

[10] U. Contractor, C. Dixit, and D. Mahajan, "CNNs for surveillance footage scene classification," 2018. [Online]. Available: arXiv:1809.02766.

[11] A. Siddique and H. Medeiros, "Tracking passengers and baggage items using multi-camera systems at security checkpoints," 2020. [Online]. Available: arXiv:2007.07924.

[12] A. Petrozziello and I. Jordanov, "Automated Deep Learning for Threat Detection in Luggage from X-Ray Images," *Lecture Notes in Computer Science*, pp. 505–512, Jun. 2019, doi: 10.1007/978-3-030-34029-2_32.

[13] J.-M. O. Steitz, F. Saeedan, and S. Roth, “Multi-view X-Ray R-CNN,” *Lecture Notes in Computer Science*, pp. 153–168, Oct. 2018, doi: 10.1007/978-3-030-12939-2_12.

[14] P. Lázaro and A. Maiorano, “Image recognition for x-ray luggage scanners using free and open source software,” 2017. [Online]. Available: http://sedici.unlp.edu.ar/handle/10915/63489

[15] N. Jaccard, T. W. Rogers, E. J. Morton and L. D. Griffin, "Automated detection of smuggled high-risk security threats using deep learning," *7th International Conference on Imaging for Crime Detection and Prevention (ICDP 2016)*, Madrid, Spain, 2016, pp. 1-6, doi: 10.1049/ic.2016.0079.

[16] A. F. Otoom, H. Gunes and M. Piccardi, "Automatic Classification of Abandoned Objects for Surveillance of Public Premises," *2008 Congress on Image and Signal Processing*, Sanya, China, 2008, pp. 542-549, doi: 10.1109/CISP.2008.688.

[17] J. Cavada, C. E. Cortés, and P. A. Rey, “A simulation approach to modelling baggage handling systems at an international airport,” *Simulation Modelling Practice and Theory*, vol. 75, pp. 146–164, Jun. 2017, doi: 10.1016/j.simpat.2017.01.006.

[18] C. Malandri, M. Briccoli, L. Mantecchini, and F. Paganelli, “A Discrete Event Simulation Model for Inbound Baggage Handling,” *Transportation Research Procedia*, vol. 35, pp. 295–304, Jan. 2018, doi: 10.1016/j.trpro.2018.12.008.

[19] D. Hafilah, A. Cakravastia, Y. Lafdail, and N. Rakoto, “Modeling and Simulation of Air France Baggage Handling System with Colored Petri Nets,” *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2443–2448, Jan. 2019, doi: 10.1016/j.ifacol.2019.11.573.

[20] OpenCV team, “OpenCV,” OpenCV.org. https://opencv.org (accessed Sep. 28, 2022)

[21] M. Abadi et al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” *arXiv (Cornell University)*, Jan. 2015, [Online]. Available: http://datascienceassn.org/sites/default/files/TensorFlow%20-%20Large-Scale%20Machine%20Learning%20on%20Heterogeneous%20Distributed%20Systems.pdf

[22] M. A. Chandra and S. Bedi, “Survey on SVM and their application in image classification,” *International Journal of Information Technology*, vol. 13, no. 5, pp. 1–11, Oct. 2021, doi: 10.1007/s41870-017-0080-1.

[23] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi and S. Homayouni, "Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 6308-6325, 2020, doi: 10.1109/JSTARS.2020.3026724.

[24] R. Azhar, D. Tuwohingide, D. Kamudi, Sarimuddin, and N. Suciati, “Batik Image Classification Using SIFT Feature Extraction, Bag of Features and Support Vector Machine,” *Procedia Computer Science*, vol. 72, pp. 24–30, Jan. 2015, doi: 10.1016/j.procs.2015.12.101. 72.

[25] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *2011 International Conference on Computer Vision*, Barcelona, Spain, 2011, pp. 2564-2571, doi: 10.1109/ICCV.2011.6126544.

[26] C. Harris and M. Stephens, “A Combined Corner and Edge Detector,” *Alvey Vision Conference*, Jan. 1988, doi: 10.5244/c.2.23.

[27] OpenCV team. *OpenCV: Shi-Tomasi Corner Detector & Good Features to Track*. (2022). Accessed: Dec. 8, 2022. [Online]. Available: https://docs.opencv.org/4.x/d4/d8c/tutorial_py_shi_tomasi.html

[28] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *HAL (Le Centre Pour La Communication Scientifique Directe)*, Feb. 2011, [Online]. Available: https://hal.inria.fr/hal-00650905/document

[29] Scikit-learn developers. *sklearn.svm.SVC - scikit-learn 1.2.2 documentation*. (2023). Accessed Mar. 24, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.htm