big data and
cognitive computing

MDPI

Article

Detection and Classification of Human-Carrying Baggage Using DenseNet-161 and Fit One Cycle

Mohamed K. Ramadan [1,2,*], Aliaa A. A. Youssif [3] and Wessam H. El-Behaidy [2]

[1] College of Computer Science, Nahda University in Beni Suef, Beni Suef 62521, Egypt
[2] College of Computers and Artificial Intelligence, Helwan University, Cairo 11795, Egypt
[3] College of Computing and Information Technology, Arab Academy for Science, Technology & Maritime Transport (AASTMT), Smart Village 12577, Egypt
[*] Correspondence: khaled.abotyra@nub.edu.eg

Abstract: In recent decades, the crime rate has significantly increased. As a result, the automatic video monitoring system has become increasingly important for researchers in computer vision. A person's baggage classification is essential in knowing who has abandoned baggage. This paper proposes a model for classifying humans carrying baggage. Two approaches are used for comparison using a deep learning technique. The first approach is based on categorizing human-containing image regions as either with or without baggage. The second approach classifies human-containing image regions based on the human position direction attribute. The proposed model is based on the pretrained DenseNet-161 architecture. It uses a "fit-one-cycle policy" strategy to reduce the training time and achieve better accuracy. The Fastai framework is used for implementation due to its super computational ability, simple workflow, and unique data cleansing functionalities. Our proposed model was experimentally validated, and the results show that the process is sufficiently precise, faster, and outperforms the existing methods. We achieved an accuracy of between 96% and 98.75% for the binary classification and 96.67% and 98.33% for the multi-class classification. For multi-class classification, the datasets, such as PETA, INRIA, ILIDS, and MSMT17, are re-annotated with one's direction information about one's stance to test the suggested approach's efficacy.

Keywords: deep learning; human-carrying baggage classification; transfer learning; fit one cycle policy; direction attribute; video surveillance

Citation: Ramadan, M.K.; Youssif, A.A.A.; El-Behaidy, W.H. Detection and Classification of Human-Carrying Baggage Using DenseNet-161 and Fit One Cycle. *Big Data Cogn. Comput.* **2022**, *6*, 108. https://doi.org/10.3390/bdcc6040108

Academic Editors: Dilbag Singh, Heung-No Lee and Vijay Kumar

Received: 7 September 2022
Accepted: 28 September 2022
Published: 6 October 2022

1. Introduction

Crime has expanded drastically, posing a threat to human life and property. These circumstances often occur when explosive chemicals are transported in carry-on or unclaimed baggage. Hence, detecting humans carrying baggage is a critical issue for preventing theft, identifying criminal conduct, and preventing bombings in an intelligent surveillance system.

Detecting humans carrying baggage can be accomplished conceptually by studying their looks. This approach might be a preliminary step in establishing the abandoned baggage owner [1]. The detection system monitors the threat and alerts the security team by deploying cameras in public areas such as buses, airports, railway stations, building lobbies, schools, etc. These public areas potentially impact numerous intelligent surveillance systems that require knowledge management and integration [2,3].

The detection technique faces many difficulties due to lighting circumstances, complicated backgrounds, articulate positions, and outside scenarios [4]. Moreover, one of the big problems is feature extraction. Several features were extracted, including histograms of oriented gradients (HOG) [5], local binary pattern (LBP) [6], scale-invariant feature transform (SIFT) [7], and gray-level co-occurrence matrix (GLCM) [8]. However, the two most common ways to obtain information about images are SIFT and GLCM. In addition, several machine learning techniques have been used in the classification step, such as support vector machines [9] and random forest [10].

Convolutional neural networks (CNN) [11] or deep learning techniques have recently demonstrated a significant and distinct ability to extract image features automatically. They are a solution to the challenge of extracting image features [12,13]. Consequently, they have been applied and have demonstrated success in various fields, including computer vision [14,15]. CNN tries to identify patterns in images by applying several convolutions. These convolutions can learn simple features such as lines and diagonals in the first few layers and then integrate these pieces in subsequent layers to learn deeper features. Using the patterns discovered in the previous layers, the models learn significant structures such as doors, arms, cats, and dogs in the final layer.

However, training deep architectures with millions of parameters by applying the random weights initialization technique can take days and weeks. A vast amount of data and powerful computer hardware (GPUs) are necessary to train CNN from scratch. "Transfer learning" is frequently employed to resolve these issues [16]. A CNN model is trained on a massive dataset, such as ImageNet, and becomes a pretrained model. The features learned by this pretrained model are transferred to the new model. In the transfer learning technique, the fully connected layer of the model is erased, and the remaining layers of the architecture are employed as a feature extractor for the new job. Thus, only the dense layers of the proposed model are trained.

Furthermore, setting hyperparameters is a significant challenge in CNN [17] that faces any researcher and requires many years of experience to tune these parameters [18]. The solution is the fit-one-cycle policy, a technique for reducing training time, increasing performance, and adjusting all hyperparameters of deep learning models, such as learning rate and weight decay [19]. Hence, the fit-one-cycle policy yields training outcomes superior to those obtained using the usual learning rate.

The following is a succinct summary of the contributions of this work:

1. We exploited pretrained models such as DenseNet-161, which were first applied to the problem of classifying humans who are carrying bags. We built a reliable model that can classify humans carrying bags in binary and multiclass classification. The binary classification classifies humans as either carrying baggage or not. The multiclass classification classifies humans into six classes depending on the human's viewing region: front-view, back-view, and side-view with or without carrying baggage in each view.
2. We manually re-annotated the datasets with direct information about human posture to evaluate the multi-classification performance of the proposed model. This will be essential in determining the type of bag to use in future work.
3. We used the new fit-one-cycle policy method to reduce the number of epochs and iterations in the model, allowing it to be used with large-scale data.
4. We performed many experimental validations of the proposed model across all public datasets. Our model outperformed the other methods in the literature, as demonstrated by the results.

The remaining sections of the paper are structured as follows. In Section 2, we will provide a literature review. In Section 3, we will describe the proposed model. Then, experimental results and discussions will be presented in Section 4. Finally, the conclusion and future work will be discussed in Section 5.

2. Related Works

Carried-object detection has become a subject of interest in recent years. Consequently, there is a growing interest in detecting objects carried in video sequences. Several approaches have been proposed in the literature to detect carried baggage.

The Backpack by Haritaoglu et al. [20] was one of the subject's earliest treatments. It detects carried objects by analyzing the symmetry and mobility of a person's body. It is based on the presumption that the human body is symmetrical around a vertical axis and that the limbs exhibit periodic motion when the individual is walking without objects. A temporal template dynamic model is created by matching and averaging a person's foreground blobs in brief video sequences. Next, the non-symmetric regions of the temporal templates are separated by computing a vertical symmetric axis using principal component analysis (PCA) and analyzing the distance between the silhouette pixels and the body axis. The non-symmetric regions are subjected to periodicity analysis to differentiate between limb and object-carrying regions.

Javed et al. [21] noted that calculating the central axis using PCA is not an effective tool for estimating the vertical symmetry axis because the shape of the silhouette is warped. Thus, the symmetry axis shifts when a person carries an object. They propose that the vertical line passing through the head's centroid is a more suitable choice for the axis of symmetry. The recurrent motion image (RMI) feature vector has been introduced to estimate the recurrent motion of a person's body. Non-symmetric regions exhibiting periodic signals in the RMI are identified as limbs, while the other regions are categorized as carried objects. As the silhouette shape represents the human object class, several studies have explored the use of prior information about the human body shape to distinguish it within a foreground blob and to elucidate the remaining foreground aspects in terms of carried objects.

Using a dynamic shape model of human motion, Lee et al. [22] detected carried objects as outliers in the extracted foreground using a dynamic shape model. Using kinematic manifold embedding and kernel mapping, they represent the dynamic shape deformations of various individuals from different viewpoints. Comparing a person's silhouette with the best-matching dynamic shape model reveals that carried objects are mismatched.

In Chayanurak et al. [23], a star skeleton represents the human silhouette. The authors discovered individuals carrying objects by analyzing a time series of extracted skeletal limb movements. Tracked limbs that stay still or move with the rest of the body are called "carried objects".

Tzanidou and Edirisinghe [24] suggested a strategy for detecting and classifying baggage by predicting the direction of motion of a human that is carrying baggage and aligning a temporal human-like template with the best-matched view-specific exemplars. They used a classification scheme based on human body parameters with significant drawbacks to classify baggage.

Shahbano et al. [25] created a method for carrying baggage detection and classification by utilizing a scalable histogram of oriented gradients (SHOG) and joint scale local binary patterns (JSLBP) to extract features of human parts and baggage. SHOG achieved high precision by permitting the selection of highly discriminative features and achieved an accuracy score of 95.4% on binary classification using INRIA, ILIDS, and MSMT17 datasets [26–28]. Then, Shahbano et al. [29] proposed a novel technique by integrating the three-dimensional link between pixels and applying a local tridirectional pattern descriptor to acquire information about the local intensities and produce an accurate feature description and size and achieved an accuracy score of 95% on binary classification using INRIA and MSMT17 datasets [26,28].

Damen et al. [30] proposed a technique for detecting objects in a human region based on constructing a spatial-temporal template from a series of human areas while walking. The template was matched with view-specific offline examples to obtain the optimal match. As a carried object, the temporal protrusion between them was acknowledged. Tzanidou et al. [31] augmented the method used in [30] by integrating color information to precisely pinpoint and move an object's location. However, the technique presupposes that those parts of carried objects extend beyond body silhouettes. Due to its dependence on the protrusion, the method cannot detect carried objects that do not protrude.

Wahyono et al. [32] devised a human-baggage detector to overcome the protruding issue by modeling the human region as several body parts, including the head, torso, leg, and baggage. Using a mixed model, carried object detection was accomplished by combining feature extraction and training on each component and merging them. Without expecting the carried object to be a protrusion part, Ghadiri et al. [33] provided a collection of contour exemplars of humans in various standing and walking positions. The carrying object is identified by studying the contour alignment between the hypothesis mask and contour exemplars. However, the approaches rely primarily on human detection outcomes, rendering them impractical since we must construct both human and object detectors.

Recently, Wahyono et al. [34] used custom CNN layers, which have proven to be practical classification applications. Moreover, they proposed a framework for classifying human-carrying objects that can be directly applied to all possible candidate regions. It is based on a convolutional neural network with a transfer learning strategy using the human viewing direction attribute. It achieved an average F1 score of 0.91 on multiclass classification using the PETA [35] dataset.

From previous literature, we found that the pretrained models were not used for classifying human-carrying objects. However, the results of the pretrained models outperformed the custom CNN layers [36]. As a result, we will build a classification model for humans carrying baggage using the DenseNet-161 pretrained model and the fit-one-cycle policy method for binary and multiclass classification.

3. Proposed Model

This section discusses the design of the proposed model, shown in Figure 1. Input images for datasets are processed in advance, as indicated in the following subsection. Then, they are passed to the training phase, as clarified in the second subsection. The last subsection describes the evaluation criteria used to assess the model.

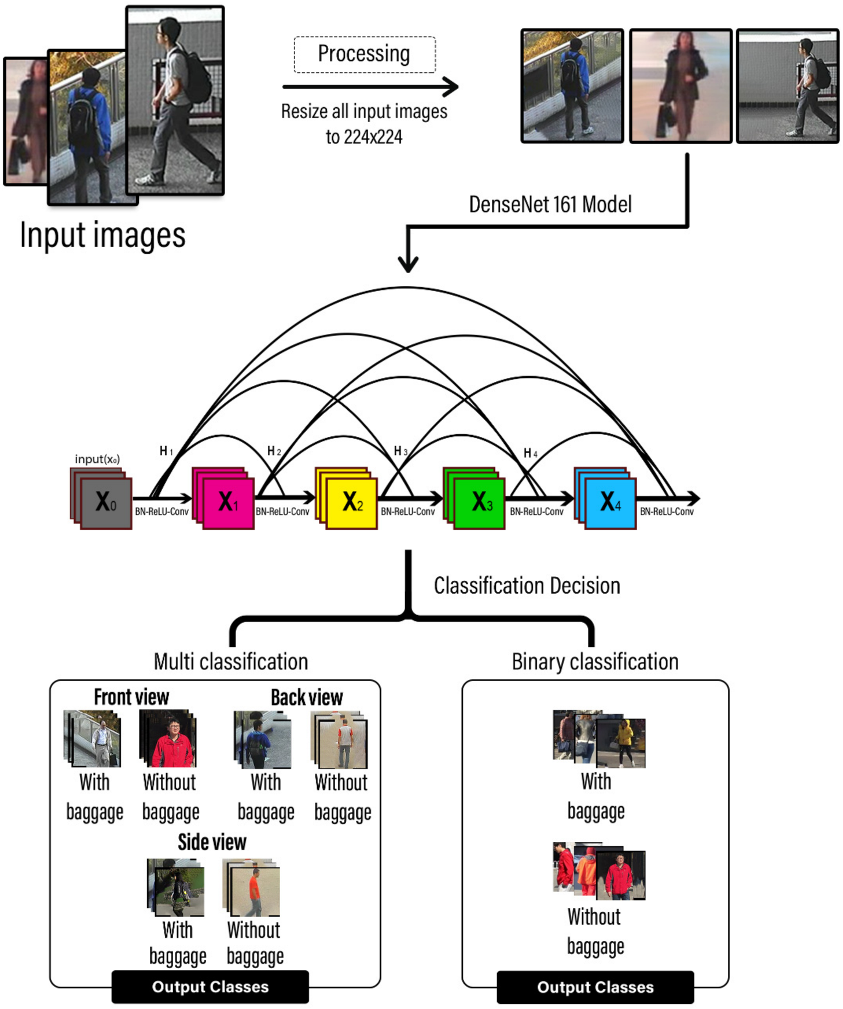


Figure 1. The processing steps in our suggested model's approach.

3.1. Data Preprocessing

This subsection illustrates the preprocessing performed on different datasets for re-annotation and resizing.

3.1.1. Re-annotation datasets

Since the camera is positioned at a fixed angle and location, the candidate regions may comprise individuals from several viewing orientations, either front, back, or side views. Thus, the viewing direction of a person is considered part of the target. Using human features, such as viewing order, has improved the accuracy of person re-identification [37,38]. We extend the number of targets for multi-class classification to six different classes, namely, front, back, and side views with baggage, and front, back, and side views without baggage. In Section 4, we will show several samples of images of humans in different views from all the datasets used in the experiment.

3.1.2. Resize Images

As a result of using many datasets, such as PETA, INRA, ILIDS, and MSMT17 that have different image sizes, all the input images are processed by resizing them to 224 × 224 pixels to fit the input of our model.

3.2. Training Phase

This subsection discusses our training phase. Firstly, we present the DenseNet-161 architecture used for our model. Then, we discuss the training method to optimize our model's run-time and performance.

3.2.1. DenseNet-161 Architecture

It is essential to highlight that the proposed model is based on the pretrained DenseNet-161 architecture.

Huang et al. [39] created dense convolutional networks (DenseNet) that achieved the greatest classification performance in 2017 on massive datasets such as ImageNet and CIFAR-10. ResNets acquire duplicate feature maps and have many parameters, making training challenging. Instead, DenseNet contains very thin layers and learns a small number of feature maps, where each layer feeds information to adjacent layers. Figure 2 illustrates the connecting mechanism of multiple convolutional layers in a dense block of DenseNet. These feed-forward connections raise the total number of layers from L to L (L+1)/2. As a result, the network's training becomes more efficient, overfitting is reduced, and the model's performance improves. DenseNet-161 consists of four dense blocks with (6, 12, 36, 24) sub-blocks, respectively. Each sub-block consists of two convolution layers, which leads to a total of 156 layers (in dense block) plus five convolution layers with a growth rate of k = 48, where k is referred to as feature maps. Figure 3 depicts a block diagram of DenseNet-161. The transition layer plays a role in model compression. If the number of channels of the output feature map from the dense block module is m, we allow the following transition layer to construct [θm] output feature maps, where $0 < \theta \leq 1$ is referred to as the compression factor. When θ = 1, the number of feature mappings across transition layers remained constant. The model is suitable for embedded devices because of its thick connectivity and compressibility.

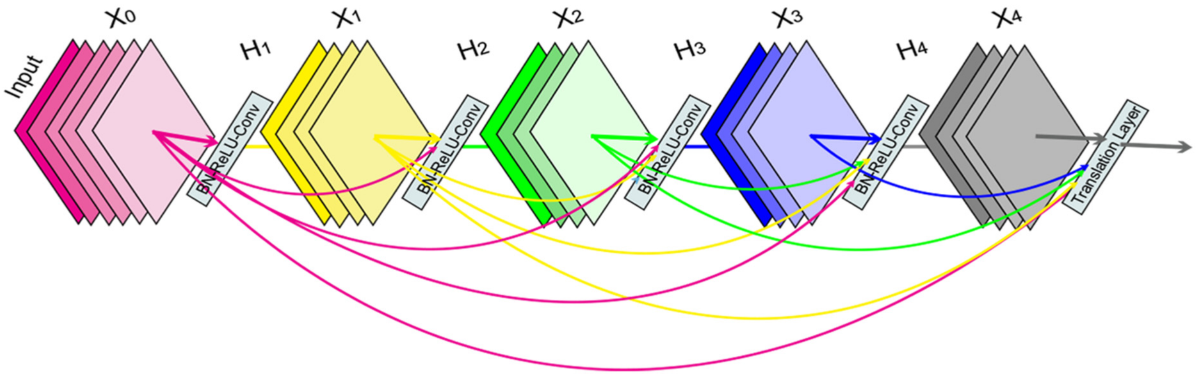


Figure 2. A dense block that illustrates that each layer takes all the previous feature maps as input.

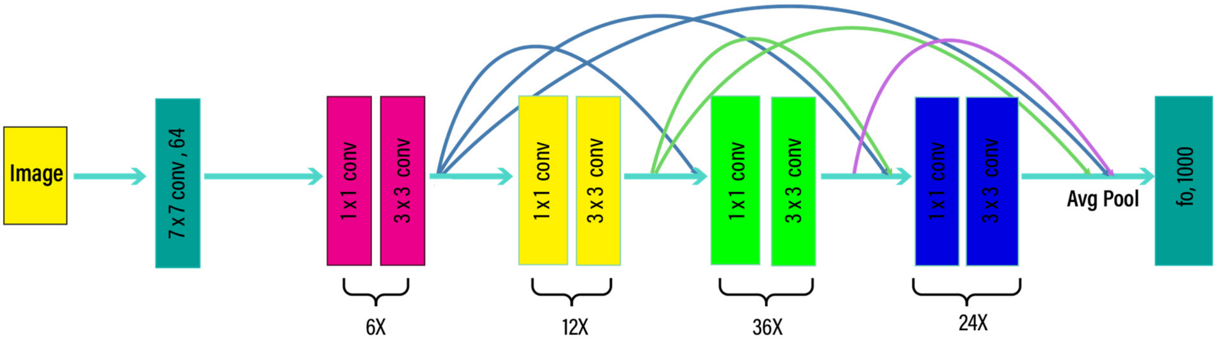


Figure 3. A block diagram illustrating the pretrained DenseNet-161 model.

Using DenseNet-161 as a pretrained model, we transferred the knowledge (weights value) of the fundamental structures learned in the first and middle layers to the proposed model. The essential parameters, that pretrained models have learned to classify various objects in the ImageNet dataset, are used to classify images of humans carrying baggage. Therefore, transfer learning accelerates the training process and the development of new CNN models.

Figure 1 explains applying DenseNet-161 as the final stage in obtaining the binary and multiclass classification results. Still, the last "fully connected" layer is modified according to our proposed model from 1000 classes to 2 classes for binary classification and from 1000 types to 6 classes for multi-classification. DenseNet-161 is applied and trained using the fit-one-cycle method.

3.2.2. The Fit-One-Cycle Method

Tuning hyper-parameters such as learning rate (LR) and momentum is crucial. There is a problem with a high learning rate; the optimizer might increase training loss rather than decrease it. Having a low learning rate not only slows down the training process but can also lead to training errors that are difficult to correct. To find the correct LR, you must conduct numerous trials and be patient.

Smith [19] established a new way of setting the learning rate called the fit-one-cycle method, which is used in this paper. The training starts with a specified learning rate and momentum. During the first half of the training, the learning rate is increased up to a fixed maximum value while the momentum is decreased. Near the end of the training, the learning rate is reduced while the momentum is increased. This strategy yields more stable results and requires fewer epochs to train our model to completion

Smith's article "*Super-Convergence*" [40] demonstrated the improvement in validation accuracy when comparing the standard learning rate policy and the one-cycle policy. In Smith's article, we saw that the training accuracy changes quickly as the learning rate (LR) goes up, oscillates when the learning rate is very high, and then jumps to a very high level of accuracy.

3.3. Evaluation Criteria

The evaluation criteria are used to evaluate the classification model's performance, including accuracy, macro-F1, micro-F1, precision, recall/sensitivity, and specificity. As shown in Equation (1), accuracy takes the sum of true positive and negative elements as the numerator and the sum of all the confusion matrix entries as the denominator. Equation (2) defines precision by dividing the total number of true positives across all classes by the total number of true positives and false positives across all classes. In Equation (3), recall/sensitivity is calculated by dividing the total number of true positives across all classes by the total number of true positives and false negatives across all classes. Equation (4) defines the F1-score as the harmonic average between precision and recall. Macro-F1 is calculated by averaging the precision and recall of each type. Micro-F1 is the sum of all the classes' true positives, false positives, and negatives. In Equation (5), specificity is calculated by dividing the total number of true negatives by the sum of true negatives and false negatives.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall/Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F1} - \text{Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{4}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{5}$$

4. Results and Discussion

In this section, we first describe the collection of data used. Next, the experimental setting of our proposed model is presented. Then, the various results of our experiments are mentioned for binary and multi-class classification. Finally, a comparison is made between our model's outcomes and the previous literature's consequences.

4.1. Dataset Description

The datasets used in our research are the PETA [35], INRIA [26], ILIDS [27], and MSMT17 [28] datasets. We used a part of each dataset that included people with and without baggage, taken from different camera orientations with direct information regarding one's position in each dataset. Then, these images are divided into two classes for binary classification and six for multi-classification. Therefore, the datasets were manually re-annotated to suit our experiments. Figure 4 shows samples from each dataset used in the experiment.

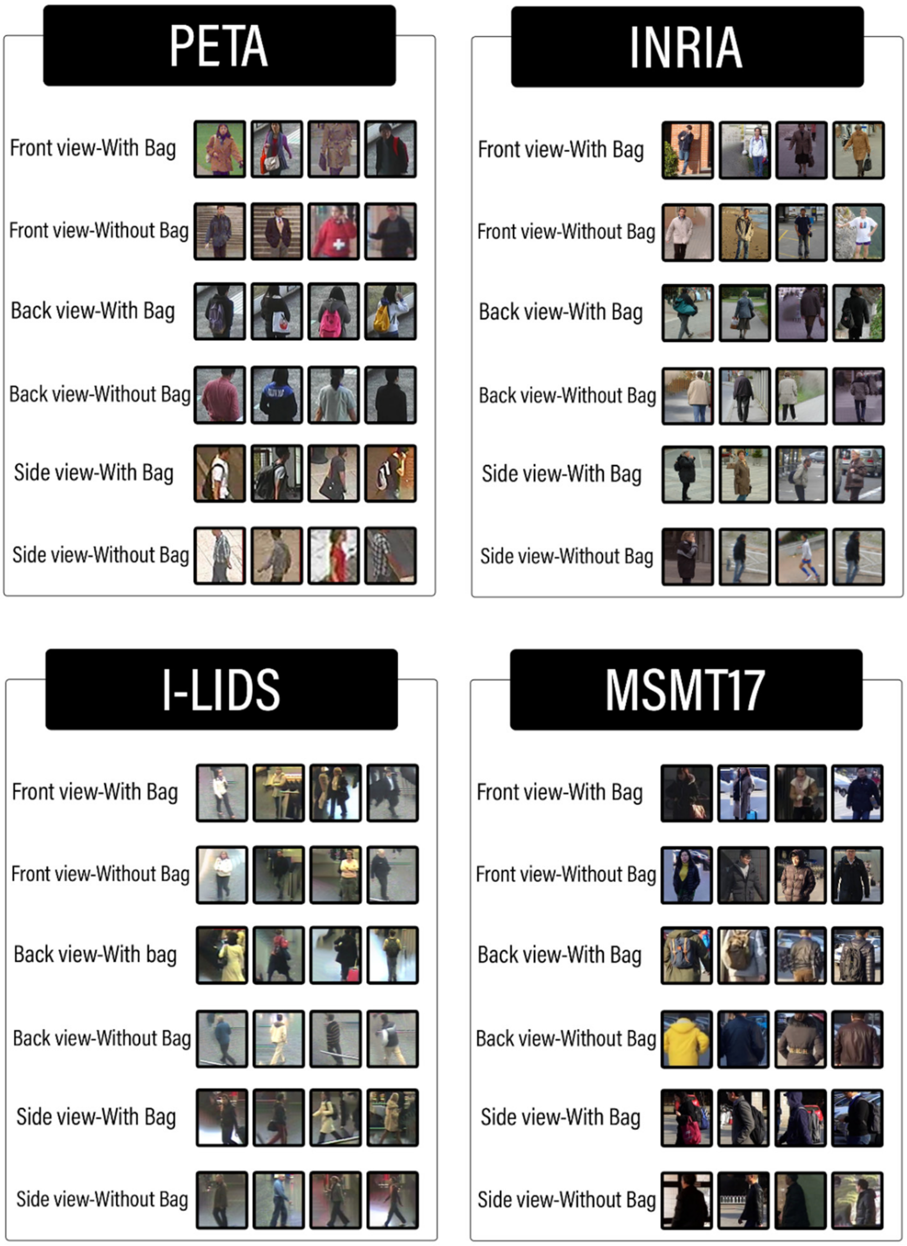


Figure 4. Images from the PETA, INRA, ILIDS, and MSMT17 databases are used as examples from each dataset. This figure demonstrates how one's status might vary throughout the six classes.

4.1.1. PETA Dataset

The PETA [35] dataset consists of 19,000 images ranging in resolution from 17-by-39 to 169-by-365 pixels. These images represent 8705 individuals. We used 6000 images: 1000 images in each class, as shown in Table 1.

Table 1. PETA dataset distribution.

Viewing Direction	Carrying Baggage	Without Baggage
Front view	1000	1000
Back view	1000	1000
Side view	1000	1000
Total	3000	3000

4.1.2. INRIA Dataset

The INRIA [26] person dataset is a collection of images of individuals used for pedestrian detection research and consists of 902 person detections. We used 112 images: 67 of individuals carrying baggage and 45 not carrying baggage, as shown in Table 2.

Table 2. INRA dataset distribution.

Viewing Direction	Carrying Baggage	Without Baggage
Front view	26	19
Back view	22	15
Side view	19	11
Total	67	45

4.1.3. ILIDS Dataset

The ILIDS [27] dataset is a human re-identification dataset featuring 300 different pedestrians observed from two unique camera perspectives in public open spaces. It comprises 600 images of 300 unique individuals and a set of image sequences from two camera views for each person. There are 43,060 images overall in the dataset. We used 4688 images, as shown in Table 3.

Table 3. ILIDS dataset distribution.

Viewing Direction	Carrying Baggage	Without Baggage
Front view	809	773
Back view	788	766
Side view	781	771
Total	2378	2310

4.1.4. MSMT17 Dataset

MSMT17 [28] is a multi-scene and multi-time person re-identification dataset. The dataset includes 180 hours of video collected by 12 outdoor and three indoor cameras during 12 intervals. It contains 126,441 image sequences of 4101 different individuals. We used 6600 images: 3451 images for carrying baggage and 3149 not carrying baggage, as shown in Table 4.

Table 4. MSMT17 dataset distribution.

Viewing Direction	Carrying Baggage	Without Baggage
Front view	1264	1046
Back view	1147	1101
Side view	1040	1002
Total	3451	3149

4.2. Experimental Settings

The Fastai framework [41] was used for model construction. It is a cutting-edge PyTorch-based framework primarily used for object identification, image segmentation, and image classification. It provides faster computations than competitors and includes data cleansing widgets, offering a highly user-friendly workflow and simplifying the debugging process. In addition, we performed our experiments using Google Colab [42].

4.3. Classification Results

Four datasets are used to test the proposed model and compare our results with the literature. Some experiments used the same number of sample images used in the literature for comparison. The other experiments used 1000 sample images per class in binary or multiclass classification, which is the recommended rule of thumb [43].

In all experiments, the datasets are randomly divided into a 65% training set, a 15% validation set, and a 20% testing set. For binary classification, the model is trained for 40 epochs, and for multi-classification, it is trained for 60 epochs with a 32-batch size for both.

4.3.1. Binary Classification

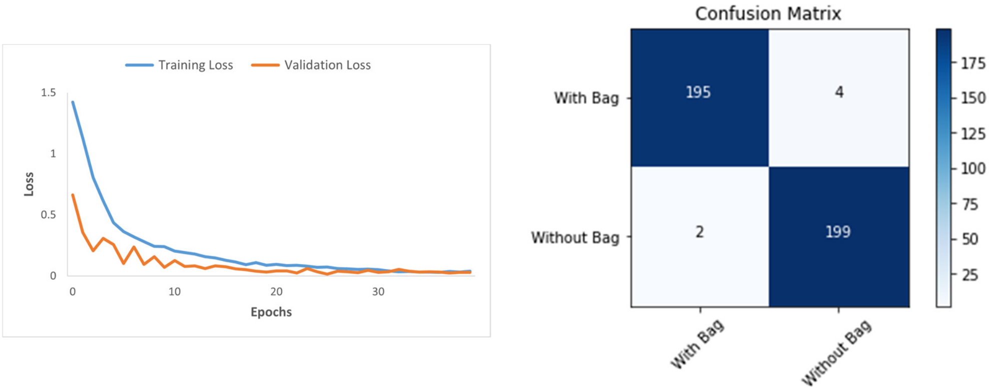
The suggested approach for binary classification was initially analyzed to categorize human-containing image regions as either with or without baggage. The samples that contain carry baggage are positive, while those that do not have carry baggage are negative samples.

PETA Dataset Results

Using our model, we used 1000 positive and 1000 negative sample images to categorize areas. All the classification results are listed in Table 5, and the obtained accuracy is 98.5%. Figure 5 shows the loss curve and the confusion matrix of the experiment.

Table 5. The evaluation results on the PETA dataset.

Network	Accuracy	Precision	Recall/Sensitivity	Specificity
Dense-Net 161	98.50%	97.99%	98.98%	98.03%



(**a**) (**b**)

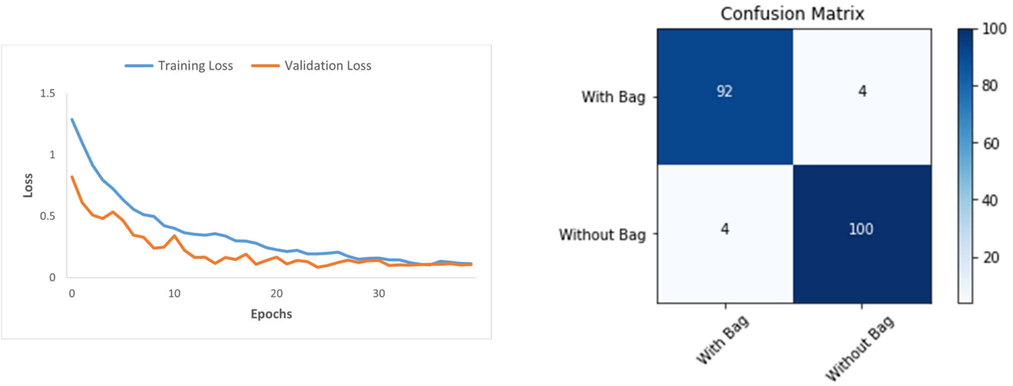
Figure 5. The loss curve and confusion matrix of binary classification on the PETA dataset: (**a**) The loss curve, (**b**) the confusion matrix.

INRIA and MSMT17 Datasets Results

This experiment used part of the INRIA and MSMT17 datasets. We conducted two experiments with different sample images. The first experiment uses 500 positive and 500 negative sample images. All the classification results are listed in Table 6, and the obtained accuracy is 96%. Figure 6 shows the loss curve and the confusion matrix of the experiment.

Table 6. The evaluation results of the first experiment on INRIA and MSMT17 datasets.

Network	Accuracy	Precision	Recall/Sensitivity	Specificity
Dense-Net 161	96.00%	95.83%	95.83%	96.15%



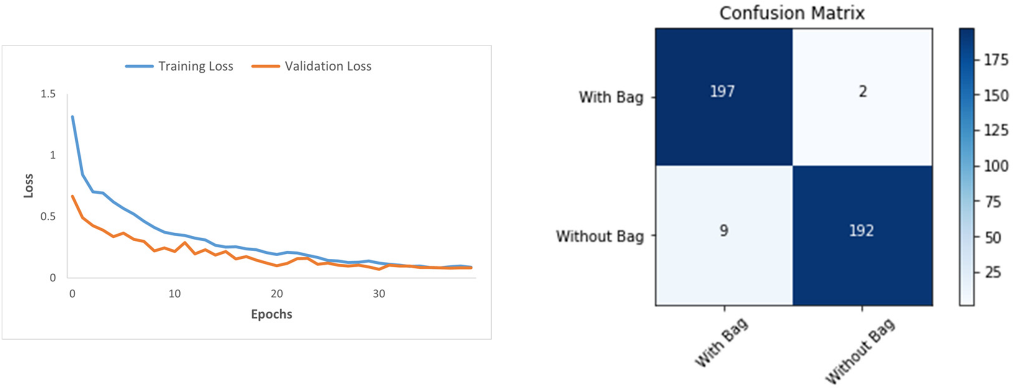
(**a**) (**b**)

Figure 6. The loss curve and confusion matrix of the first experiment on INRIA and MSMT17 datasets: (**a**) The loss curve, (**b**) the confusion matrix.

In the second experiment, we used 1000 positive sample images and 1000 negative sample images as the rule of thumb. All the classification results are given in Table 7, and the obtained accuracy is 97.25%. The loss curve and the confusion matrix are shown in Figure 7.

Table 7. The evaluation results of the second experiment on INRIA and MSMT17 datasets.

Network	Accuracy	Precision	Recall/Sensitivity	Specificity
Dense-Net 161	97.25%	98.99%	95.63%	98.97%



(**a**) (**b**)

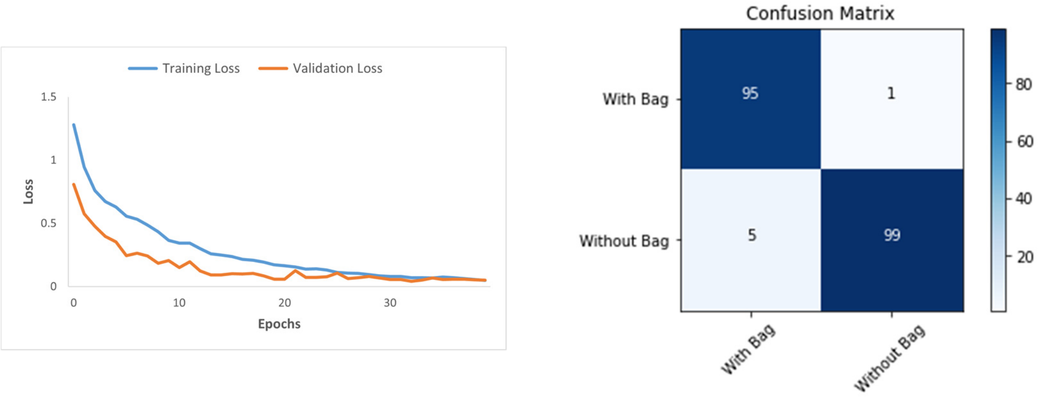
Figure 7. The loss curve and confusion matrix of the second experiment on INRIA and MSMT17 datasets: (**a**) The loss curve, (**b**) the confusion matrix.

INRIA, ILIDS, and MSMT17 Datasets Results

This experiment used part of the INRIA, ILIDS, and MSMT17 datasets. We conducted two experiments with different sample images. The first experiment uses 500 positive and 500 negative sample images. All the classification results are listed in Table 8, and the obtained accuracy is 97%. Figure 8 shows the loss curve and the confusion matrix of the experiment.

Table 8. The evaluation results of the first experiment on INRIA, ILIDS, and MSMT17 datasets.

Network	Accuracy	Precision	Recall/Sensitivity	Specificity
Dense-Net 161	97.00%	98.96%	95.00%	99.00%



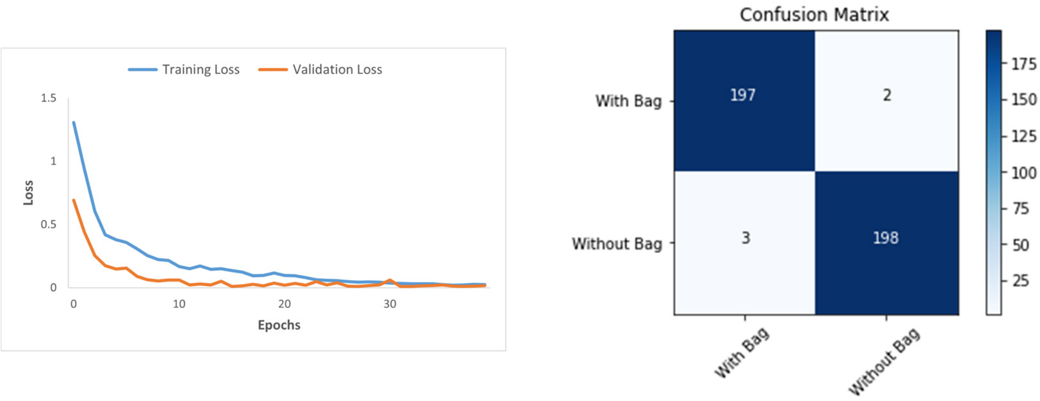
(**a**) (**b**)

Figure 8. The loss curve and confusion matrix of the first experiment on INRIA, ILIDS, and MSMT17 datasets: (**a**) The loss curve, (**b**) the confusion matrix.

In the second experiment, we used 1000 positive sample images and 1000 negative sample images as the rule of thumb. All the classification results are given in Table 9, and the obtained accuracy is 98.75%. The loss curve and the confusion matrix are shown in Figure 9.

Table 9. The evaluation results of the second experiment on INRIA, ILIDS, and MSMT17 datasets.

Network	Accuracy	Precision	Recall/Sensitivity	Specificity
Dense-Net 161	98.75%	98.99%	98.50%	99.00%



(**a**) (**b**)

Figure 9. The loss curve and confusion matrix of the second experiment on INRIA, ILIDS, and MSMT17 datasets: (**a**) The loss curve, (**b**) the confusion matrix.

4.3.2. Multi Classification

The suggested approach for multi-classification was analyzed to categorize human-containing image regions based on the human position direction attribute (for example, front, back, and side views) as either with or without baggage. The samples containing a carrying bag with any direction attribute are positive, while those that do not have a carrying bag with any direction attribute are negative.

PETA Dataset Results

Using our model based on the human position direction attribute, we used part of the PETA dataset to categorize areas. In the first experiment, we chose sample images of 2673 (same as in [34]), distributed as shown in Table 10. All the classification results are listed in Table 11, and the obtained accuracy is 97%. Table 12 shows precision, recall, and F1 scores, whereas Figure 10 shows the experiment's loss curve and confusion matrix.

Table 10. PETA dataset distribution for multi-classification for the first experiment.

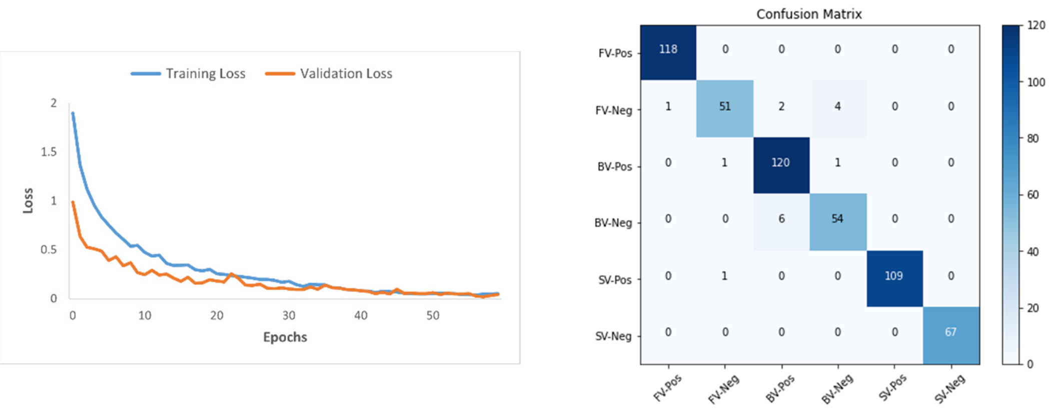
Viewing Direction	Carrying Baggage	Without Baggage
Front view	330	562
Back view	318	605
Side view	325	533
Total	973	1700

Table 11. The evaluation results of the first experiment on the PETA dataset.

Network	Accuracy	Macro F1	Micro F1
Dense-Net 161	97%	96.29%	97%

Table 12. Precision, recall, and F1-score for the first experiment on the PETA dataset.

	Precision	Recall	F1-Score
FV-Pos	1.00	0.99	1.00
FV-Neg	0.88	0.96	0.92
BV-Pos	0.98	0.94	0.96
BV-Neg	0.90	0.92	0.91
SV-Pos	0.99	1.00	1.00
SV-Neg	1.00	1.00	1.00
Average	0.96	0.97	0.97



(**a**) (**b**)

Figure 10. The loss curve and confusion matrix of the first experiment on the PETA dataset: (**a**) The loss curve, (**b**) the confusion matrix.

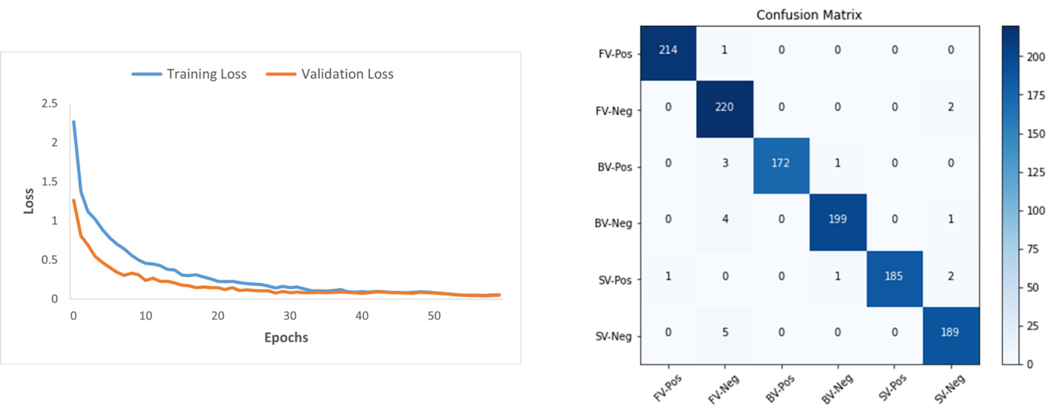
In the second experiment, we used 6000 images (1000 per class) as the rule of thumb. All the classification results are listed in Table 13, and the obtained accuracy is 98.25%. Table 14 shows precision, recall, and F1 scores. Figure 11 shows the loss curve and the confusion matrix of the experiment.

Table 13. The evaluation results of the second experiment on the PETA dataset.

Network	Accuracy	Macro F1	Micro F1
Dense-Net 161	98.25%	98.28%	98.25%

Table 14. Precision, recall, and F1-score for the second experiment on the PETA dataset.

	Precision	Recall	F1-Score
FV-Pos	1.00	1.00	1.00
FV-Neg	0.99	0.94	0.97
BV-Pos	0.98	1.00	0.99
BV-Neg	0.98	0.99	0.98
SV-Pos	0.98	1.00	0.99
SV-Neg	0.97	0.97	0.97
Average	0.983	0.983	0.983



(**a**) (**b**)

Figure 11. The loss curve and confusion matrix of the second experiment on the PETA dataset: (**a**) The loss curve, (**b**) the confusion matrix.

INRIA and MSMT17 Datasets Results

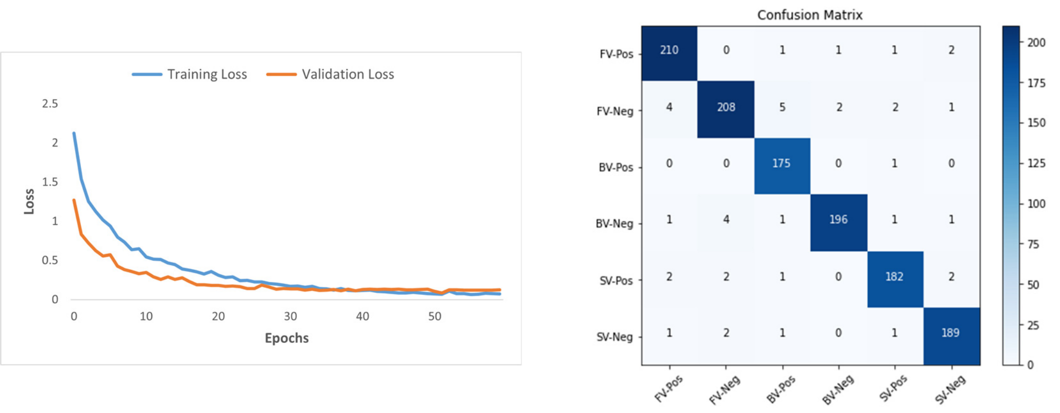
We used a part of the INRIA and MSMT17 datasets. We used 6000 images, 1000 for each class. All the classification results are listed in Table 15, and the obtained accuracy is 96.67%. Table 16 shows precision, recall, and F1 scores. Figure 12 shows the loss curve and the confusion matrix of the experiment.

Table 15. The evaluation results on INRIA and MSMT17 datasets.

Network	Accuracy	Macro F1	Micro F1
Dense-Net 161	96.67%	96.69%	96.67%

Table 16. Precision, recall, and F1-score for INRIA and MSMT17 datasets.

	Precision	Recall	F1-Score
FV-Pos	0.98	0.96	0.97
FV-Neg	0.94	0.96	0.95
BV-Pos	0.99	0.95	0.97
BV-Neg	0.96	0.98	0.97
SV-Pos	0.96	0.97	0.97
SV-Neg	0.97	0.97	0.97
Average	0.967	0.965	0.967



	FV-Pos	FV-Neg	BV-Pos	BV-Neg	SV-Pos	SV-Neg
FV-Pos	210	0	1	1	1	2
FV-Neg	4	208	5	2	2	1
BV-Pos	0	0	175	0	1	0
BV-Neg	1	4	1	196	1	1
SV-Pos	2	2	1	0	182	2
SV-Neg	1	2	1	0	1	189

(**a**) (**b**)

Figure 12. The loss curve and confusion matrix of multi-classification on INRIA and MSMT17 datasets: (**a**) The loss curve, (**b**) the confusion matrix.

INRIA, ILIDS, and MSMT17 Datasets Results

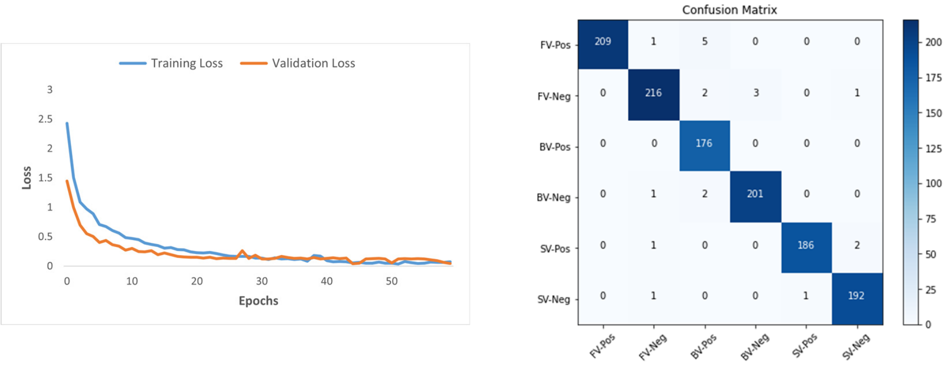
We used a part of the INRIA, ILIDS, and MSMT17 datasets. We used 6000 images, 1000 for each class. All the classification results are listed in Table 17, and the obtained accuracy is 98.33%. Table 18 shows precision, recall, and F1 scores. Figure 13 shows the loss curve and the confusion matrix of the experiment.

Table 17. The evaluation results on INRIA, ILIDS, and MSMT17 datasets.

Network	Accuracy	Macro F1	Micro F1
Dense-Net 161	98.33%	98.26%	98.33%

Table 18. Precision, recall, and F1-score for INRIA, ILIDS, and MSMT17 datasets.

	Precision	Recall	F1-Score
FV-Pos	0.97	1.00	0.99
FV-Neg	0.97	0.98	0.98
BV-Pos	1.00	0.95	0.98
BV-Neg	0.99	0.99	0.99
SV-Pos	0.98	0.99	0.99
SV-Neg	0.99	0.98	0.99
Average	0.983	0.982	0.987



	FV-Pos	FV-Neg	BV-Pos	BV-Neg	SV-Pos	SV-Neg
FV-Pos	209	1	5	0	0	0
FV-Neg	0	216	2	3	0	1
BV-Pos	0	0	176	0	0	0
BV-Neg	0	1	2	201	0	0
SV-Pos	0	1	0	0	186	2
SV-Neg	0	1	0	0	1	192

(**a**) (**b**)

Figure 13. The loss curve and confusion matrix of multi-classification on INRIA, ILIDS, and MSMT17 datasets: (**a**) The loss curve, (**b**) the confusion matrix.

4.4. A Comparison between the Proposed Model and the Existing Models

This section compares the results of our model based on DenseNet-161 with the literature on different datasets. Accuracy, precision, recall/sensitivity, and specificity are terms of comparison for binary classification. In contrast, accuracy and the average of these metrics (precision, recall, F1-score) are terms of difference for multi-classification. The comparison of binary classification is shown in Table 19, and the comparison of multi-classification is shown in Table 20. The evaluation of the results of our model indicates that it achieved the best results by applying a method in training called fit-one-cycle with a small number of batches and the fewest number of epochs. This helped us reduce the training time and achieve better accuracy than the alternative research techniques described in our research paper.

Table 19. Comparison of binary classification.

Dataset	Number of Images	Method	Accuracy	Precision	Recall/Sensitivity	Specificity
INRIA and MSMT17	500 per class	Our Method DenseNet-161	96% *	0.9583	0.9583 *	0.9615 *
	1000 image per class		97.25%	0.9899	0.9563	0.9897
	500 per class	Local Tridirectional Pattern [29]	95%	0.9889	0.9511	0.8333
INRIA, ILIDS, and MSMT17	500 per class	Our Method DenseNet-161	97% *	0.9896 *	0.9500	0.9900 *
	1000 image per class		98.75%	0.9899	0.9850	0.9900
	500 per class	Joint Scale LBP [25]	95.4%	0.9889	0.9511	0.8333
PETA	1000 images per class	Our Method DenseNet-161	98.50%	0.9799	0.9898	0.9803

Note: The numbers with "*" in the table are the best results by comparing the same number of images with other methods in the literature.

Table 20. Comparison for multi-classification.

Dataset	Number of Images	Method	Accuracy	Av. Precision	Av. Recall	Av. F1-Score
PETA	2673 images	Our Method DenseNet-161	97%	0.96	0.97	0.97
	1000 per class		98%	0.98	0.98	0.98
	2673 images	CNNR + DA + TL [34]	–	0.93	0.90	0.91
	2673 images	CNNR [34]	–	0.94	0.64	0.76
	2673 images	CNN + BA + TL [34]	–	0.95	0.68	0.79
	2673 images	CNN + DA + RF [34]	–	0.90	0.69	0.78
	2673 images	CNN + BA + SVM [34]	–	0.86	0.50	0.63
	2673 images	CNN [34]	–	0.93	0.55	0.70
INRIA and MSMT17	1000 per class	Our Method DenseNet-161	96.67%	0.97	0.97	0.97
INRIA, ILIDS, and MSMT17			98.33%	0.98	0.98	0.99

4.4.1. Comparison of Binary Classifications

We have compared our model based on DenseNet-161 against two models from the literature [25,29] for classifying humans carrying baggage. We recorded the experimental results in Table 19. The first row of Table 19 shows the results of our model on INRIA and MSMT17 datasets, in addition to the results of recent work by Shahbano et al. [29] using the local tridirectional pattern method. These results show that our model achieved better accuracy by 1% using the same number of images and by 2.25% using 1000 images per class. In the second row of Table 19, we show the results of our model on INRIA, ILIDS, and MSMT17 datasets, and the results of Shahbano et al. [25] using the joint scale LBP method. These results show that our model achieved better accuracy by 1.6% using almost the same number of images and by 3.1% using 1000 images per class. To ensure the effectiveness of our model on different datasets with different variations in illumination and resolution, the third row of Table 19 shows the results of our model on the PETA dataset. These results indicate that our model achieved an accuracy rate of 98.5% using 1000 images per class.

By comparing the same number of images, our model is better on almost all measures (asterisk results). However, the results for 1000 images are the best of all models.

4.4.2. Comparison of Multi Classifications

We have compared our model based on Densenet-161 against the six methods of CNN experimented with by Wahyono et al. [34] for classifying humans carrying baggage. We recorded the experimental results in Table 20. The first row of Table 20 shows the results of our model on the PETA dataset, in addition to the results of [34], using various methods. Among the six methods, "CNNR + DA + TL" has the highest F1 score, which [34] considers the best method. Our model achieved a better average for these metrics (precision, recall, and F1-score) than the best method by 3% for precision, 7% for recall, and 6% for F1-score using the same number of images. In addition, our model achieved a better average than the best method by 5% for precision, 8% for recall, and 7% for F1 score by using 1000 images per class, and to ensure the effectiveness of our model on different datasets by using 1000 images per class. The second row of Table 20 shows the results of our model on the INRIA and MSMT17 datasets. It achieves 0.97 for the average F1 score. Furthermore, the third row shows the results for the INRIA, ILIDS, and MSMT17 datasets, as it reaches 0.99 for the average F1 score.

Based on these results, we conclude that our model is reliable over different datasets, including various challenges.

5. Conclusions and Future Work

This research proposed a model for classifying people carrying baggage. We employed pretrained DenseNet-161 to perform a reliable model for binary classification and multi-classification. In addition, this proposed model utilizes a fit-one-cycle to reduce the number of cycles required to train the CNN and enhance the proposed model's accuracy. Additionally, to assess the multi-classification performance of the proposed model, we re-annotated the datasets used with direct information regarding one's position. The experiments were performed on the PETA, INRA, ILIDS, and MSMT17 datasets, and the results were evaluated using several performance metrics. After extensive experiments on different datasets, the proposed model for binary classification showed a classification that reached an accuracy rate of 98.75%. Additionally, multi-classification shows a classification that reached an accuracy rate of 98.33%. We observed that binary classification gives better accuracy than multiple classifications, which is normal due to the increased number of classes. However, multi-classification techniques can handle baggage detection when baggage is heavily obscured, and objects cannot be visually distinguished. It can also be used as a starting point to determine the type of bag a person is carrying based on viewing direction attributes.

Author Contributions: Conceptualization, A.A.A.Y. and W.H.E.-B.; data curation, M.K.R.; formal analysis, M.K.R. and A.A.A.Y.; investigation, W.H.E.-B.; methodology, M.K.R.; supervision, A.A.A.Y.; validation, W.H.E.-B.; writing—original draft, M.K.R.; writing—review and editing, A.A.A.Y. and W.H.E.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to the reason that we use the public datasets, and the figures with humans are from the public datasets.

Informed Consent Statement: Informed consent was waived for this study due to the reason that we use the public datasets and because the figures with humans are from the public datasets.

Data Availability Statement: PETA Dataset (http://mmlab.ie.cuhk.edu.hk/projects/PETA.html, accessed on 16 February 2022), INRIA Dataset (https://paperswithcode.com/dataset/inria-person, accessed on 19 November 2021), ILIDS Dataset (https://xiatian-zhu.github.io/downloads_qmul_iLIDS-VID_ReID_dataset.html, accessed on 26 March 2022), MSMT17 Dataset (http://www.pkuvmc.com/dataset.html/, accessed on 23 February 2022).

Acknowledgments: The authors are grateful to the reviewers and editors for providing advice and guidance as to how to improve the paper.

Conflicts of Interest: All authors declare that they have no conflicts of interest.

References

1. Filonenko, A.; Jo, K.-H. Unattended Object Identification for Intelligent Surveillance Systems Using Sequence of Dual Background Difference. *IEEE Trans. Ind. Inform.* **2016**, *12*, 2247–2255.
2. Altunay, D.G.; Karademir, N.; Topçu, O.; Direkoğlu, C. Intelligent Surveillance System for Abandoned Luggage. In Proceedings of the 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2–5 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–4.
3. Budiharto, W.; Gunawan, A.A.S.; Suroso, J.S.; Chowanda, A.; Patrik, A.; Utama, G. Fast Object Detection for Quadcopter Drone Using Deep Learning. In Proceedings of the 2018 3rd International Conference on Computer and Communication Systems (ICCCS), Nagoya, Japan, 27–30 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 192–195.
4. Yanagisawa, H.; Yamashita, T.; Watanabe, H. A Study on Object Detection Method from Manga Images Using CNN. In Proceedings of the 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand, 7–9 January 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–4.
5. Surasak, T.; Takahiro, I.; Cheng, C.; Wang, C.; Sheng, P. Histogram of Oriented Gradients for Human Detection in Video. In Proceedings of the 2018 5th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand, 17–18 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 172–176.
6. Satpathy, A.; Jiang, X.; Eng, H.-L. Human Detection Using Discriminative and Robust Local Binary Pattern. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 2376–2380.
7. Lowe, D.G. Object Recognition from Local Scale-Invariant Features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; IEEE: Piscataway, NJ, USA, 1999; Volume 2, pp. 1150–1157.
8. Haralick, R.M.; Shanmugam, K.; Dinstein, I.H. Textural Features for Image Classification. *IEEE Trans. Syst. Man. Cybern.* **1973**, *SMC-3*, 610–621. [CrossRef]
9. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
10. Ho, T.K. Random Decision Forests. In Proceedings of the of 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; IEEE: Piscataway, NJ, USA, 1995; Volume 1, pp. 278–282.
11. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
13. Xie, J.; Hou, Q.; Shi, Y.; Peng, L.; Jing, L.; Zhuang, F.; Zhang, J.; Tang, X.; Xu, S. The Automatic Identification of Butterfly Species. *arXiv* **2018**, arXiv:1803.06626.
14. Gulshan, V.; Peng, L.; Coram, M.; Stumpe, M.C.; Wu, D.; Narayanaswamy, A.; Venugopalan, S.; Widner, K.; Madams, T.; Cuadros, J. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA* **2016**, *316*, 2402–2410. [CrossRef] [PubMed]
15. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks. *Nature* **2017**, *542*, 115–118. [CrossRef] [PubMed]
16. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
17. Jastrzębski, S.; Arpit, D.; Ballas, N.; Verma, V.; Che, T.; Bengio, Y. Residual Connections Encourage Iterative Inference. *arXiv* **2017**, arXiv:1710.04773.
18. Smith, L.N. A Disciplined Approach to Neural Network Hyper-Parameters: Part 1—Learning Rate, Batch Size, Momentum, and Weight Decay. *arXiv* **2018**, arXiv:1803.09820.
19. Smith, L.N. Cyclical Learning Rates for Training Neural Networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 464–472.

20. Haritaoglu, I.; Cutler, R.; Harwood, D.; Davis, L.S. Backpack: Detection of People Carrying Objects Using Silhouettes. *Comput. Vis. Image Underst.* **2001**, *81*, 385–397. [CrossRef]
21. Javed, O.; Shah, M. Tracking and Object Classification for Automated Surveillance. In *Computer Vision, Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, 28–31 May 2002*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 343–357.
22. Lee, C.-S.; Elgammal, A. Carrying Object Detection Using Pose Preserving Dynamic Shape Models. In *Articulated Motion and Deformable Objects, Proceedings of the 4th International Conference on Articulated Motion and Deformable Objects, Port d'Andratx, Mallorca, Spain, 11–14 July 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 315–325.
23. Chayanurak, R.; Cooharojananone, N.; Satoh, S.; Lipikorn, R. Carried Object Detection Using Star Skeleton with Adaptive Centroid and Time Series Graph. In Proceedings of the IEEE 10th International Conference on Signal Processing Proceedings, Beijing, China, 24–28 October 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 736–739.
24. Tzanidou, G.; Edirisinghe, E.A. Automatic Baggage Detection and Classification. In Proceedings of the 2011 11th International Conference on Intelligent Systems Design and Applications, Cordoba, Spain, 22–24 November 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 825–830.
25. Shahbano, W.A.; Shah, S.M.A.; Ashfaq, M. Carried Baggage Detection and Classification Using Joint Scale LBP. In Proceedings of the 2020 5th International Electrical Engineering Conference (IEEC 2020) IEP Centre, Karachi, Pakistan, 21–22 February 2020.
26. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 1, pp. 886–893.
27. Li, M.; Zhu, X.; Gong, S. Unsupervised Person Re-Identification by Deep Learning Tracklet Association. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 737–753.
28. Wei, L.; Zhang, S.; Gao, W.; Tian, Q. Person Transfer Gan to Bridge Domain Gap for Person Re-Identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 79–88.
29. Shahbano, M.A.; Inayat, K. Robust Baggage Detection and Classification Based on Local Tri-Directional Pattern. *arXiv* **2020**, arXiv:2006.07345. [CrossRef]
30. Damen, D.; Hogg, D. Detecting Carried Objects from Sequences of Walking Pedestrians. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 1056–1067. [CrossRef] [PubMed]
31. Tzanidou, G.; Zafar, I.; Edirisinghe, E.A. Carried Object Detection in Videos Using Color Information. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1620–1631. [CrossRef]
32. Hariyono, J.; Jo, K.-H. Body Part Boosting Model for Carried Baggage Detection and Classification. *Neurocomputing* **2017**, *228*, 106–118.
33. Ghadiri, F.; Bergevin, R.; Bilodeau, G.-A. Carried Object Detection Based on an Ensemble of Contour Exemplars. In *Computer Vision, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 852–866.
34. Jo, K.-H. Human Carrying Baggage Classification Using Transfer Learning on CNN with Direction Attribute. In *Intelligent Computing Theories and Application, Proceedings of the International Conference on Intelligent Computing, Liverpool, UK, 7–10 August 2017*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 717–724.
35. Deng, Y.; Luo, P.; Loy, C.C.; Tang, X. Pedestrian Attribute Recognition at Far Distance. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 789–792.
36. Han, X.; Zhang, Z.; Ding, N.; Gu, Y.; Liu, X.; Huo, Y.; Qiu, J.; Yao, Y.; Zhang, A.; Zhang, L. Pre-Trained Models: Past, Present and Future. *AI Open* **2021**, *2*, 225–250. [CrossRef]
37. Layne, R.; Hospedales, T.M.; Gong, S. Towards Person Identification and Re-Identification with Attributes. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 402–412.
38. Lin, Y.; Zheng, L.; Zheng, Z.; Wu, Y.; Hu, Z.; Yan, C.; Yang, Y. Improving Person Re-Identification by Attribute and Identity Learning. *Pattern Recognit.* **2019**, *95*, 151–161. [CrossRef]
39. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, 21–26 July 2017; pp. 4700–4708.
40. Smith, L.N.; Topin, N. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. In Proceedings of the Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, Baltimore, MD, USA, 14–18 April 2019; SPIE: Bellingham, WA, USA, 2019; Volume 11006, pp. 369–386.
41. Howard, J.; Gugger, S. Fastai: A Layered API for Deep Learning. *Information* **2020**, *11*, 108. [CrossRef]
42. Google Colab. Available online: https://colab.research.google.com/ (accessed on 23 September 2022).
43. Shahinfar, S.; Meek, P.; Falzon, G. "How Many Images Do I Need?" Understanding How Sample Size per Class Affects Deep Learning Model Performance Metrics for Balanced Designs in Autonomous Wildlife Monitoring. *Ecol. Inform.* **2020**, *57*, 101085. [CrossRef]