Recent Advances in Business Analytics. Selected papers of the 2021 KNOWCON-NSAIS workshop on Business Analytics pp. 37–44
DOI: 10.15439/2021B4
ISSN 2300-5963 ACSIS, Vol. 29

# Image based classification of shipments using transfer learning

Markus Leppioja
markus.leppioja@gmail.com
Loihde Analytics
Valtakatu 49, 53100 Lappeenranta, Finland

Pasi Luukka
pasi.luukka@lut.fi
LUT University
School of Business & Management
Yliopistonkatu 34, 53850
Lappeenranta, Finland

Christoph Lohrmann
christoph.lohrmann@lut.fi
LUT University
School of Business & Management
Yliopistonkatu 34, 53850
Lappeenranta, Finland

***Abstract*—This paper focuses on recognizing different postal shipment types from images taken by the sorting machine. Greyscale images obtained from sorting machines are used to build a classifier using transfer learning to recognize seven different classes of shipments. Three convolutional neural networks (VGG16, GoogLeNet and ResNet50), that were pretrained using the ImageNet dataset, were used as feature extractors and the extracted features were subsequently supplied to a neural network classifier. VGG16 demonstrated the best performance for six out of the seven classes and achieved an overall mean accuracy of 95.69% on the independent test set. The model accomplished F1 scores exceeding 90% for five out of seven classes, only having a lower recall for the aggregated class "Other" and shipments from abroad. The results of this study highlight the potential of transfer learning for computer vision in the context of shipment classification.**

## I. Introduction

THE objective in this study is to build a classifier to effectively recognize different shipment types from images taken by a sorting machine. Data for the shipment type classification problem is obtained from a company operating in the field of postal and logistics services. Different types of shipments arrive from several sources to the company's networks. These shipments pass through a sorting process which divides the shipments based on the location of the destination. However, the sorting machine is not capable of recognising the type of each shipment and the number of shipments of each type, which are both of interest to the company. Especially the recognition of consumer-to-consumer letters is pivotal since there are no preannouncements related to this shipment type whereas some larger customers make preannouncements about future shipments to ensure their smooth processing. Thus, being able to recognize the type of shipments, especially the "Consumer Letter" type, but also all other types, is the main aim of this work. The problem presents itself as a computer vision problem where an image is taken by a sorting machine and a classifier needs to be built to recognize which shipment type is present in the image. From this information, the quantities for all types of shipments can be inferred, thus addressing both objectives for the case company.

For this type of problem deep learning and convolutional neural networks (CNN) have proven to be useful. Nowadays the databases that CNNs are trained on are so large that at least low-level features extracted in the first convolutional blocks are useful in almost any computer vision application. Thus, the features extracted from such pretrained models are commonly used, whereas training a new CNN from scratch is rare [1]. The advantage of using a pretrained CNN is that it is computationally less complex, and less data is needed to fit a new classifier than for fully training a CNN model. Limitations on the computational complexity are also the reason for the application of a pretrained CNN in this study.

Pretrained convolutional neural networks are usable in many different fields. For example, Pardamean et al. [2] had a small size mammogram dataset and used transfer learning of a convolutional neural network pretrained on chest X-ray data to overcome this problem. The best model was able to achieve a 90.38% accuracy. Sun and Qian [3] worked on a Chinese herbal medicine recognition task from images using a pretrained convolutional neural network VGG16. They managed to achieve an average precision of 71% which these authors considered promising. Reddy and Juliet [4] used transfer learning with the objective to classify malarial infected cells and improve the malaria diagnostics accuracy with the pretrained convolutional neural network ResNet50. They reported to have obtained an accuracy of 95.4%. In the study of Chmielinska's and Jakubowski [5] the problem was to develop a detector for driver fatigue symptoms based on facial images. Driver fatigue is considered one of the main causes for car accidents. In this case the authors used a pretrained convolutional neural network called AlexNet. Their results indicate that it is possible to use transfer learning for the detection of driver fatigue symptoms. The best class had an error rate of less than 2%. Abu Mallouh et al. [6] worked on classifying peoples' age range from images. They managed to show that pretrained CNNs can be used for this problem. Their model outperformed the previous state of the art solution by 12%. Sert and Boyacı [7] worked on a free-hand sketch recognition problem. They deployed three pretrained convolutional neural networks for feature extraction: AlexNet, VGG16 and GN-Triplet [8]. A support vector machine was used as a classifier. The model which was able to achieve the best accuracy of 97.91% used a combination of AlexNet and GN-Triplet together with PCA. Fu and Aldrich [9] used convolutional neural networks for analysing a froth flotation process from images. In their study AlexNet performed the best and managed to outperform the previous best solutions. Shao et al. [10] worked on a machine fault diagnostic problem. They selected the VGG16 pretrained convolutional neural network for their study. The best performing, finetuned VGG16 model's accuracy was reported to be almost 100%. The recognition of plant species was the sub-

ject in the research problem covered by Ghazi et al. [11]. They used three different pretrained convolutional neural networks: VGG16, AlexNet and GoogLeNet. The best performing model with accuracy of 80.18% was achieved with a combination of VGG16 and GoogLeNet. Data augmentation and finetuning the number of iterations was considered the most important factors influencing the results. Tree species identification from wooden boards was the subject in the study by Shustrov et al. [12]. They used the four convolutional neural network architectures AlexNet, VGG16, GoogLeNet and ResNet to address this problem. The highest accuracy of 94.7 % was obtained with GoogLeNet. Besides this, Camargo et al. [13] used the pretrained convolutional neural network AlexNet to classify sunspots and were able to achieve an accuracy of 91.70%. Finally, Zhao et al. [14] built a classifier for land-use with a transfer learning technique and spatial resolution images available for the land-use.

The results show that transfer learning based on pretrained convolutional neural networks was successfully applied in many different fields and contexts. It is thus also selected for the machine vision problem in this study.

## II. Convolutional Neural Networks

Fully connected neural networks connect each neuron in a layer with all neurons in the subsequent layer [15]. Since the weight of each of these connections represents a parameter to be learned during model training, fully connected neural networks tend to have a large number of parameters that need to be trained [1]. This problem is amplified when there are many neurons in each layer and / or there are many layers in the network - which is not uncommon in deep learning problems. The key idea behind convolutional neural networks (CNN) is to create a solution in a way that reduces the number of parameters compared to fully connected neural networks. This allows to train deeper networks with less parameters [16], [29], [30].

One of the first convolutional architectures was LenNet-5, which was applied to identify hand-written numbers [17]. Since LeNet-5, convolutional neural networks have evolved in terms of the number of layers and the use of different activation functions.

Convolutional neural networks are combinations of convolutional and pooling layers. The last layers are usually fully connected ones. The network can be defined through the number of filters, stride lengths, the number of convolution pooling combinations and the fully connected layers. Fig. 1 represents such a simple network [18].

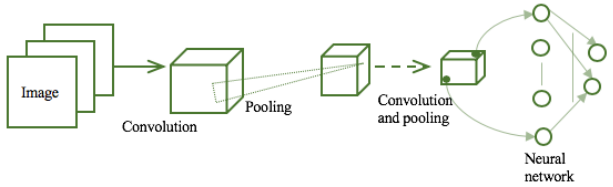


Fig 1. A simple convolutional neural network, reproduced from Rebala et al. [18]

The key aspect of convolutional neural networks is an operation called “convolution”. Convolution is a dot product operation between grid-structured inputs and a grid-structured set of weights which is drawn from different spatial localities in the input volume. It is useful when there is a high level of spatial locality in the data, for instance, in case of image data.

The goal of the pooling layer is to reduce the dimensionality of feature maps. Hence, the pooling can be called “down sampling”. In a pooling operation, the maximum (or sometimes the average) of a small grid region is returned [1]. The pooling is applied to every feature map separately, whereas a convolution operation uses all feature maps simultaneously [1], [16]. This is the reason why the pooling operation doesn’t change the number of feature maps – the depth stays the same [1]. Nevertheless, the dimensionality of the feature maps reduces spatially [16].

The convolutional neural network works in a similar way as a regular feed-forward neural network. The difference is that the operations in the layers are spatially organized with sparse connections. The ReLU activation typically follows the convolutional operation hence it is not usually shown independently when illustrating convolutional neural networks. Compared to other common activation functions, ReLU is advantageous in terms of speed and accuracy [1].

Convolutional neural networks allow translation invariance [19]. This means, for instance, in images that an object is the same object no matter where it is located in the image [19]. This is related to weight (or parameter) sharing - a particular shape should be processed the same way regardless of its spatial location [1]. There has been a great advancement in the field of image classification in the 2010s due to the development of the ImageNet database [20]. It contains over 14 million images with a large number of sub-categories [21]. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is a competition where participants use the ImageNet database in different tasks. ILSVRC has been arranged from 2010 to 2017 yearly and many state-of-the-art CNN architectures have participated and won the challenge.

### A. VGG

Visual Geometry Group’s (VGG) convolutional neural network placed second in the ILSVRCs image classification task. Simonyan and Zisserman [22] present different versions of their model in their article, for instance VGG16 and VGG19. The architecture of VGG16 is shown in Fig. 2.

There are 16 weight layers in VGG16, out of which there are 13 convolutional weight layers. In between each two to three convolutional layers is a max-pooling layer. Moreover, the three last layers are fully connected. The ReLU activation function is selected in the convolutional part and in the first two fully connected layers, while the softmax activation function is used in the last layer which provides the class probabilities (outputs). The core idea is to use 3x3 filters instead of the widely used 5x5 or 7x7 filters. In particular, a 3x3 filter is used three times in a row. The advantage of this approach is that the decision function is more discriminative. Another advantage is that there are less parameters in this approach compared to the versions with 5x5 or 7x7 fil-

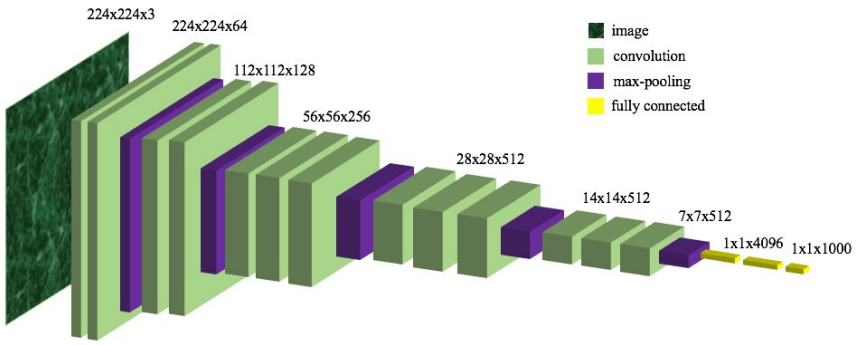


Fig 2. Illustration of the VGG16 architecture

ters, reducing the overfitting problem. There are altogether 138 million parameters in the VGG16 model [22].

### B. GoogLeNet

GoogLeNet is a convolutional neural network architecture and the winner of the ILSVRC 2014 challenge in image classification [23]. To reduce the dimensionality and the computation load, GoogleLeNet heavily relies on 1x1 convolutions. The inception module is displayed in Fig. 3. The idea of inception modules is to extract features using 1x1, 3x3, 5x5 convolutions and 3x3 max-pooling and then combine them together [24].

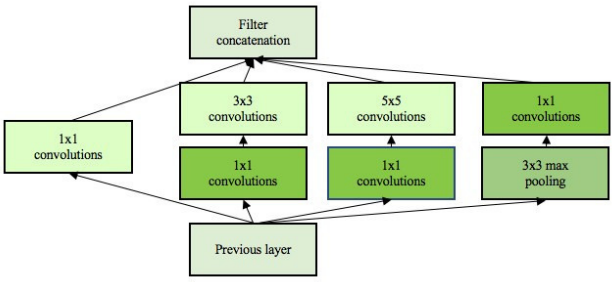


Fig 3. Inception module, reproduced from Szegedy et al. [24]

GoogLeNet is a deep CNN, containing 27 layers - counting both weight layers (22) and pooling (5) layers. All the convolutions are using the ReLU activation, also the convolutions inside the inception modules.

GoogLeNet uses one average pooling layer instead of a fully connected layer after the convolutional layers and, thus, reduces overfitting. There is also one dropout layer after the average pooling layer. The last layer is fully connected, and it uses a softmax activation. On top of the original GoogLeNet model, some of the authors have introduced modifications called InceptionV2 and InceptionV3. The goal of these modifications is to scale up the network and add regularization in as computationally efficient ways as possible [24].

### C. ResNet

ResNet is a CNN architecture and the winner of the ILSVRC 2015 image classification task. The winning model contained 152 trainable layers. It is the deepest model ever presented in the ILSVRC. However, it is noteworthy that the complexity of ResNet-152 is still lower than VGG's CNN [25]. Deep convolutional neural networks suffer from the vanishing/exploding gradient problem. This increases the error in a very deep CNN. The solution to the stated problem is shortcut connections as shown in Fig. 4. The shortcut connection can skip one or more layers and the outputs are added to the outputs of the stacked layer. This reduces the vanishing/exploding gradient problem and allows to build deeper networks. Basic identity shortcut connections do not add parameters or complexity to the model. Identity shortcuts can be used when the input and output have the same dimensions [25].

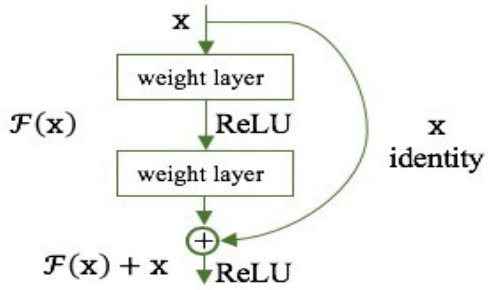


Fig 4. The identity shortcut connection, reproduced from [25]

A bottleneck design is used for the deep ResNet models. In particular, 1x1 convolutions are added to the start and the end of the network. This approach is the same kind as in GoogLeNet. Convolutional layers use the ReLU activation. After the convolutional part, the average pooling and one fully connected layer are used [25].

### D. Transfer learning and finetuning

Deep convolutional neural networks contain a large number of parameters. The large number of parameters ensures their ability to learn complex tasks. However, it also means that a considerable amount of data is needed to fully train such models adequately. Having said this, for many applications a large amount of labelled data might not be available [1], [26]. If there is not sufficient training data, the model will suffer from overfitting and won't generalize well [26]. Data availability is the key reason why the technique called transfer learning has been developed.

Pattanayak [26] (p. 211) describes transfer learning as follows: "Transfer learning in a broad sense refers to storing knowledge gained while solving a problem using that

knowledge for a different problem in a similar domain." Aggarwal [1] points out that using features extracted from public data sources, such as ImageNet, can be viewed as transfer learning. This is beneficial for image data since features extracted from a certain dataset are reusable across data sources [1]. For a new problem, less data is needed because low-level features were already extracted previously from another data domain. The reason for this is that when images are processed through many layers of convolutions, the initial layers learned to detect universal features such as shapes and edges [1], [26].

The simplest way to implement transfer learning is to remove the original output layer of an existing, trained model and replace it with the one suitable to the new problem [19]. Another option is to remove the topmost layers of the original network and use the output as features (inputs) in a new machine learning model [19]. The new machine learning model can be, for instance, a support vector machine, a random forest or a neural network [19]. There is also the possibility to freeze certain layers of the pretrained model and then retrain the model [19]. This means that weights of the frozen layers are not updated during the training [19]. Retraining some of the layers is often referred to as 'finetuning' [1], [27].

## III. Shipment Type Classification Problem and Results.

### A. Dataset and transfer learning strategy

The dataset used in this study contains images of shipments from sorting machines with the shipment ID and shipment type. The shipment type was classified manually (by hand). The size of the dataset is 25'979 shipments with 13 different shipment types. The rarest shipment types were grouped together to the class "Other", so that this classification problem eventually contained only seven different classes (Table I).

TABLE I.
Classes and their frequency of occurrence in the dataset.

| Class number | Class name | Number of samples |
|---|---|---|
| **0** | Image not found | 333 |
| **1** | Consumer letter | 1'677 |
| **2** | Commercial shipment | 3'938 |
| **3** | Shipment from abroad | 1'125 |
| **4** | Corporate letter | 16'265 |
| **5** | Magazine | 1'956 |
| **6** | Other | 685 |

To build the classifier, a technique of transfer learning is used. Three pretrained convolutional neural networks, VGG16, GoogLeNet and ResNet50 are used. All three models were selected since they are commonly used for image classification in the literature and, additionally, have demonstrated their ability to perform well on challenging image classification problems e.g., in the ILSVRC. The top layers of these models are removed, and all the other layers remain frozen. The pretrained models are used as feature extractors. On top of these models, a simple three layered fully connected neural network classifier is utilized. The first layer is a dense layer, which takes the features as an input. It contains 256 nodes and uses the ReLU activation. The next layer is a dropout layer. This is used to avoid overfitting and to add regularization. A dropout ratio of 0.5 is used. The final layer is the output layer, which makes the actual prediction. The activation function softmax is used in this layer. The output is a probability distribution. The class which has the highest probability is the one that the model predicts. Different classes are evaluated in terms of F1 score, precision as well as recall and based on the results, there is a variability of the model's performance on the different classes.

Based on the scientific literature, three different CNN architectures were selected for the application to this problem. These were the 16-layer VGG model (VGG16), GoogLeNet (InceptionV3) and the 50-layer ResNet (ResNet50). The strategy was to apply transfer learning to these models, which had been pretrained on the ImageNet dataset, and to compare these models' performance to find the best one for classifying the shipment types from images.

The data is divided into training and testing sets with a 90-10% split (holdout method). Additionally, a 10-fold cross validation is performed for the training data and the results are averaged over the 10 folds of the cross validation. The batch size is set to 25 and the number of epochs to 100.

### B. Results from the models

The classification accuracies, cross-entropy losses and standard deviations of the validation results are displayed in Table II. The highest accuracy of 95.11% was obtained with the VGG16. However, the other two models were also capable of achieving an accuracy of over 90 %. The lowest categorical cross-entropy loss was obtained by ResNet50 and the highest by VGG16. The sample standard deviation of VGG16's loss was relatively high compared to the other models. The results indicate that VGG16 might be suffering from some degree of overfitting. This was supported by the observation that the loss value varied much between the folds, compared to GoogLeNet and ResNet50. However, the model is clearly performing best in terms of accuracy.

Additionally, $F_1$ scores and their sample standard deviations are presented for each class and each model in Table III. It is noteworthy that all models tend to perform poorer on the classes "Other" and "Shipment from abroad" and also have clearly higher sample standard deviations. Overall, VGG16 produces the best $F_1$ scores in six out of seven classes.

TABLE II.
Accuracies, categorical cross-entropy losses and their sample standard deviation (validation results).

| | VGG16 | GoogLeNet | ResNet50 |
|---|---|---|---|
| **Accuracy** | **95.11%** (+-0.6265%) | 91.87% (+-0.5622%) | 93.24% (+-0.4261%) |
| **Categorical cross-entropy loss** | 1.121 (+-0.2988) | 0.3621 (+-0.0337) | 0.3773 (+-0.0341) |

In Table IV the precision and recall values together with their standard deviations are reported for VGG16. Since $F_1$ scores are based only on precision and recall, the results for

TABLE III.
F1 scores and sample standard deviations of each class, the highest F1 score for each class is in bold.

| Method | VGG16 $F_1$ score | GoogLeNet $F_1$ score | ResNet50 $F_1$ score |
|---|---|---|---|
| **Image not found (0)** | 89.47% (+-4.457%) | **89.51%** (+-3.483%) | 87.58% (+-4.640%) |
| **Consumer letter (1)** | **92.89%** (+-1.234%) | 87.74% (+-1.414%) | 90.10% (+-1.913%) |
| **Commercial shipment (2)** | **92.75%** (+-1.739%) | 86.74% (+-1.785%) | 89.54% (+-1.081%) |
| **Shipment from abroad (3)** | **83.85%** (+-3.750%) | 72.17% (+-4.434%) | 77.09% (+-4.103%) |
| **Corporate letter (4)** | **97.78%** (+-0.3716%) | 95.96% (+-0.3776%) | 96.70% (+-0.2292%) |
| **Magazine (5)** | **91.27%** (+-1.721%) | 88.43% (+-1.875%) | 89.82% (+-1.534%) |
| **Other (6)** | **79.48%** (+-5.286%) | 68.49% (+-5.723%) | 73.21% (+-6.879%) |

TABLE IV.
Precision, recall and their sample standard deviations for VGG16 (validation results).

| VGG16 | Precision | Recall |
|---|---|---|
| **Image not found (0)** | 97.36% (+-3.445%) | 83.06% (+-6.969%) |
| **Consumer letter (1)** | 93.45% (+-1.972%) | 92.38% (+-1.361%) |
| **Commercial shipment (2)** | 92.94% (+-1.866%) | 92.59% (+-2.086%) |
| **Shipment from abroad (3)** | 91.66% (+-4.002%) | 77.52% (+-5.812%) |
| **Corporate letter (4)** | 96.89% (+-0.5723%) | 98.69% (+-0.3279%) |
| **Magazine (5)** | 90.09% (+-1.827%) | 92.52% (+-2.484%) |
| **Other (6)** | 86.95% (+-5.098%) | 73.51% (+-7.310%) |

GoogleLeNet and ResNet50 are lower for most of the classes also in terms of these two metrics and can be found in Table VII and Table VIII in the appendix.

According to Table IV, all precision values for VGG16 are relatively high. Two of the lowest precision values, which are also characterized by high sample standard deviations, are linked to the "Other" and "Shipment from abroad" classes. For instance, a precision value of over 90 % was achieved for all classes, except for the class "Other".

A similar situation is encountered for the recall of VGG16, where the "Other" and "Shipment from abroad" classes both show values below 80% - the lowest recalls of all classes. On the "Consumer letter" class, which is one of the classes of the highest interest for the case company, the model is overall performing well: the precision value is 93.45% and the recall value is 92.38%.

### C. *Test set results*

Applying VGG16 on the test set, an accuracy of 95.69% and a categorical cross-entropy loss value of 0.9176 were achieved, which are close to the average results obtained on the validation sets. These results indicate that VGG16 is indeed performing well and has the ability to generalize its performance for shipment classification. The test set's $F_1$ score, precision and recall are presented in Table V. The $F_1$ score is higher than 90% for five out of seven classes and is still above 80% for the "Shipment from abroad" and "Other" classes. When compared to the validation results, it is apparent that for the "Consumer letter" class the $F_1$ score, precision and recall are a bit lower in the test set results.

The recall values of the "Shipment from abroad" and "Other" class are comparably low. The low recall values indicate that the classifier is not able to identify these classes very well from the samples and many of the samples that actually belong to these classes are falsely assigned to one of the other classes. One reason for the low recall value is that the class "Other" consists of several smaller classes which were combined to one (13 classes originally of which seven were aggregated into this class). This of course also entails that samples in this class are more dissimilar among each other than in other classes. The results indicate that this clearly has an effect on the recall (and precision) for this class. Another reason for the low recall in this class can be the low sample size. Overall, there were only 685 samples in this class which is the second smallest of all classes. The fact that the class "Image not found", which has the smallest sample size but is not aggregated, has a considerably lower recall than all other classes (other than "Other" and "Shipment from abroad") reinforces this reasoning.

TABLE V.
F1 score, precision and recall for each class of the test set.

| VGG16 | $F_1$ score | Precision | Recall |
|---|---|---|---|
| **Image not found (0)** | 90.14% | 96.97% | 84.21% |
| **Consumer letter (1)** | 91.93% | 92.25% | 91.61% |
| **Commercial shipment (2)** | 94.01% | 93.42% | 94.62% |
| **Shipment from abroad (3)** | 83.81% | 92.63% | 76.52% |
| **Corporate letter (4)** | 98.02% | 96.98% | 99.09% |
| **Magazine (5)** | 93.99% | 94.24% | 93.75% |
| **Other (6)** | 80.65% | 90.91% | 72.46% |

For the class "Shipment from abroad" the comparably low performance values can be explained by the fact that it – even though it was not aggregated from classes - also contains different types of shipments, which are all coming from abroad. These shipments can vary considerably, and it seems that the classifier has some difficulty in finding the similarities between shipments belonging to this class (see Table VI). Table VI highlights that the class "Shipment from abroad" is most often misclassified into the classes "Consumer letter" and "Corporate letter".

The reason for this can be that shipments coming from abroad are often letter type shipments – making it hard to differentiate the "Shipment from abroad" class from these other two classes and, to some smaller degree, vice versa.

A noticeable misclassification error can also be detected between the classes "Commercial shipment" and "Magazine". This appears plausible since some magazines have commercial contents on the back cover. Besides this, "Commercial shipment" is a relatively heterogeneous class since it contains different kinds of shipments. Overall, the test set indicates that the classifier is performing well for the shipment type classification. Moreover, the confusion matrix and the misclassification errors are consistent with those obtained during the validation (see Table IX in the appendix).

## IV. Conclusions

In this study, pretrained convolutional neural networks were applied for a shipment type recognition problem. The convolutional neural networks were pretrained using the ImageNet dataset and a transfer learning strategy that is suitable for shipment type classification was developed. In particular, three different models were selected for the application to this particular problem: VGG16, GoogLeNet and ResNet50.

These models were used as feature extractors and the extracted features were subsequently supplied to the classifier. The classifier developed for this purpose was a simple neural network. The dataset available for this study contained images of shipments taken by sorting machines and differentiates seven classes of shipments. The highest mean accuracy of 95.11% was obtained with VGG16 selected as the feature extractor on the validation data. ResNet50 achieved a mean accuracy of 93.24% and GoogLeNet of 91.87%. For the validation data sets VGG16 performed overall the best and produced the best results in every class except one. From the business perspective, the most important class to recognise in this study was "Consumer letter". The model demonstrated on this class its second-best performance of all classes in terms of the $F_1$ score (92.89%) and precision (93.45%) and a comparably high recall (92.38%). On the independent test set, VGG16 obtained an accuracy of 95.69%, which is almost identical to the mean accuracy obtained on the validation data sets. Moreover, given that the majority class accounts for only 62.61% of the data, this result seems overall very promising. The $F_1$ score for the "Consumer letter" class in the test set was with 91.93% also comparable to that obtained during the validation. Overall, the confusion matrix also indicated that the misclassification error is largely based on plausible misclassifications that are linked to same classes being similar to each other and/or heterogenous within (e.g., "Shipments from Abroad" with "Consumer Letter" and "Corporate Letter").

It is noteworthy that there was more variability for the categorical cross-entropy loss and accuracy for the cross validated results of VGG16 in terms of the sample standard deviations than for the other models. It should be kept in mind, that the trained classifier with the VGG16 model possesses considerably more parameters than the other two models due to the larger output vector of VGG16. Because of this, there is a larger possibility to run into overfitting problems with VGG16. When for a dataset of given size, the number of parameters is larger, there is a greater chance to tune also the less useful parameters' values as part of the final model. However, given the consistently high and similar results of VGG16 for cross-validation and the test set, this is likely neither a major concern nor critical. The training of all three models was relatively fast – which is one of the main advantages of the transfer learning approach. Unsurprisingly, VGG16 took the longest to train since it has more parameters than the two other classifiers. However, training a full model from scratch would have taken considerably longer. ˇ

TABLE VI.
Confusion matrix of VGG16 of the test set.

| True labels \ Predictions | VGG 16 (n = 2598) | Image not found (0) | Consumer letter (1) | Commercial shipment (2) | Shipment from abroad (3) | Corporate letter (4) | Magazine (5) | Other (6) |
|---|---|---|---|---|---|---|---|---|
| | Image not found (0) | 32 | 0 | 3 | 0 | 3 | 0 | 0 |
| | Consumer letter (1) | 0 | 131 | 2 | 4 | 5 | 0 | 1 |
| | Commercial shipment (2) | 0 | 0 | 369 | 0 | 13 | 8 | 0 |
| | Shipment from abroad (3) | 0 | 9 | 4 | 88 | 14 | 0 | 0 |
| | Corporate letter (4) | 0 | 1 | 7 | 2 | 1636 | 1 | 4 |
| | Magazine (5) | 0 | 0 | 10 | 1 | 1 | 180 | 0 |
| | Other (6) | 1 | 1 | 0 | 0 | 15 | 2 | 50 |

## Acknowledgment

The authors acknowledge that this paper is based on Markus Leppioja's master's thesis titled "Shipment type classification from images" [28]. The authors would like to thank Artur Vuorimaa for his help in editing the text.

## References

[1] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Cham: Springer International Publishing, 2018.

[2] B. Pardamean, T. W. Cenggoro, R. Rahutomo, A. Budiarto, and E. K. Karuppiah, "Transfer Learning from Chest X-Ray Pre-trained Convolutional Neural Network for Learning Mammogram Data,"

*Procedia Comput. Sci.*, vol. 135, pp. 400–407, 2018, doi: https://doi.org/10.1016/j.procs.2018.08.190.
[3] X. Sun and H. Qian, "Chinese Herbal Medicine Image Recognition and Retrieval by Convolutional Neural Network," *PLoS One*, vol. 11, no. 6, pp. 1–19, 2016, doi: 10.1371/journal.pone.0156327.
[4] A. S. B. Reddy and D. S. Juliet, "Transfer Learning with ResNet-50 for Malaria Cell-Image Classification," in *2019 International Conference on Communication and Signal Processing (ICCSP)*, 2019, pp. 945–949, doi: 10.1109/ICCSP.2019.8697909.
[5] J. Chmielinska and J. Jakubowski, "Detection of driver fatigue symptoms using transfer learning," *Bull. Polish Acad. Sci.*, vol. 66, no. 6, pp. 869–874, 2018, doi: 10.24425/bpas.2018.125934.
[6] A. Abu Mallouh, Z. Qawaqneh, and B. D. Barkana, "Utilizing CNNs and transfer learning of pre-trained models for age range classification from unconstrained face images," *Image Vis. Comput.*, vol. 88, pp. 41–51, 2019, doi: https://doi.org/10.1016/j.imavis.2019.05.001.
[7] M. Sert and E. Boyaci, "Sketch Recognition Using Transfer Learning," *Multimed. Tools Appl.*, vol. 78, no. 12, pp. 17095–17112, 2019, doi: 10.1007/s11042-018-7067-1.
[8] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The Sketchy Database: Learning to Retrieve Badly Drawn Bunnies," *ACM Trans. Graph.*, vol. 35, no. 4, 2016, doi: 10.1145/2897824.2925954.
[9] Y. Fu and C. Aldrich, "Froth image analysis by use of transfer learning and convolutional neural networks," *Miner. Eng.*, vol. 115, pp. 68–78, 2018, doi: https://doi.org/10.1016/j.mineng.2017.10.005.
[10] S. Shao, S. McAleer, R. Yan, and P. Baldi, "Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning," *IEEE Trans. Ind. Informatics*, vol. 15, no. 4, pp. 2446–2455, 2019, doi:10.1109/TII.2018.2864759.
[11] M. Mehdipour Ghazi, B. Yanikoglu, and E. Aptoula, "Plant identification using deep neural networks via optimization of transfer learning parameters," *Neurocomputing*, vol. 235, pp. 228–235, 2017, doi: https://doi.org/10.1016/j.neucom.2017.01.018.
[12] D. Shustrov, T. Eerola, L. Lensu, H. Kälviäinen, and H. Haario, "Fine-Grained Wood Species Identification Using Convolutional Neural Networks," in *Image Analysis*, 2019, pp. 67–77.
[13] T. O. Camargo *et al.*, "Detecting a predefined solar spot group with a pretrained convolutional neural network," in *2019 IEEE Colombian Conference on Applications in Computational Intelligence (ColCACI)*, 2019, pp. 1–6, doi: 10.1109/ColCACI.2019.8781990.
[14] B. Zhao, B. Huang, and Y. Zhong, "Transfer Learning With Fully Pretrained Deep Convolution Networks for Land-Use Classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 9, pp. 1436–1440, 2017, doi: 10.1109/LGRS.2017.2691013.
[15] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer ScienceBusiness Media, 2006.
[16] H. H. Aghdam and E. J. Heravi, *Guide to convolutional neural networks. A practical application to traffic-sign detection and classification*. Springer International Publishing, 2017.
[17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.
[18] G. Rebala, A. Ravi, and S. Churiwala, *An introduction to machine learning*. Springer International Publishing, 2019.
[19] M. Salvaris, D. Dean, and W. H. Tok, *Deep learning with Azure. Building and deploying artificial intelligence solutions on the Microsoft AI platform*. Apress, 2018.
[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.
[21] ImageNet, "Summary and Statistics," 2020. http://image-net.org/about-stats (accessed Feb. 29, 2020).
[22] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations, (ICLR)*, 2015, pp. 1–14, [Online]. Available: http://arxiv.org/abs/1409.1556.
[23] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.
[24] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[26] S. Pattanayak, *Pro deep learning with TensorFlow. A mathematical approach to advanced artificial intelligence in Python*. Apress, 2017.
[27] L. Mou and Z. Jin, *Tree-Based Convolutional Neural Networks: Principles and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2018.
[28] M. Leppioja, Shipment type classification from images, Master's thesis, LUT University, 2020
[29] K. Danilchenko and M. Segal, An efficient connected swarm deployment via deep learning, Annals of Computer Science and Information Systems, 25, 2021, pp. 1-7.
[30] A. M. Nguyen and H.S. Nguyen, Rotation Invariance in Graph Convolutional Networks, Annals of Computer Science and Information Systems, 25, 2021, pp. 81-90.

## APPENDIX

TABLE VII.
Precision, recall and their sample standard deviations for GoogLeNet.(validation results)

| GoogLeNet | Precision | Recall |
|---|---|---|
| **Image not found (0)** | 97.40% (+-3.799%) | 83.09% (+-6.022%) |
| **Consumer letter (1)** | 87.97% (+-3.000%) | 87.62% (+-2.434%) |
| **Commercial shipment (2)** | 88.03% (+-3.011%) | 85.62% (+-3.264%) |
| **Shipment from abroad (3)** | 85.03% (+-5.892%) | 63.27% (+-7.008%) |
| **Corporate letter (4)** | 94.46% (+-0.8449%) | 97.52% (+-0.5742%) |
| **Magazine (5)** | 86.74% (+-3.205%) | 90.36% (+-3.545%) |
| **Other (6)** | 80.94% (+-3.453%) | 59.93% (+-8.844%) |

TABLE VIII.
Precision, recall and their sample standard deviations for ResNet50 (validation set results).

| ResNet50 | Precision | Recall |
|---|---|---|
| **Image not found (0)** | 95.11% (+-3.213%) | 81.66% (+-8.461%) |
| **Consumer letter (1)** | 91.16% (+-1.892%) | 89.11% (+-2.856%) |
| **Commercial shipment (2)** | 90.57% (+-1.658%) | 88.58% (+-2.186%) |
| **Shipment from abroad (3)** | 84.28% (+-4.302%) | 71.19% (+-5.331%) |
| **Corporate letter (4)** | 95.45% (+-0.4993%) | 97.98% (+-0.5020%) |
| **Magazine (5)** | 88.44% (+-3.589%) | 91.44% (+-3.084%) |
| **Other (6)** | 85.50% (+-7.973%) | 64.76% (+-9.042%) |

TABLE IX.
Confusion matrix of VGG16 (Average validation performance).

| | | Predictions | | | | | | |
|---|---|---|---|---|---|---|---|---|
| True labels | **VGG 16** (n = 2338.1) | **Image not found (0)** | **Consumer letter (1)** | **Commercial shipment (2)** | **Shipment from abroad (3)** | **Corporate letter (4)** | **Magazine (5)** | **Other (6)** |
| | **Image not found (0)** | 24.50 (+-2.07) | 0.30 (+-0.48) | 1.40 (+-1.17) | 0.00 (+-0.00) | 2.40 (+-1.84) | 0.30 (+-0.48) | 0.60 (+-0.84) |
| | **Consumer letter (1)** | 0.00 (+-0.00) | 141.70 (+-1.89) | 1.50 (+-1.51) | 2.80 (+-1.81) | 7.20 (+-1.93) | 0.10 (+-0.32) | 0.10 (+-0.2) |
| | **Commercial shipment (2)** | 0.10 (+-0.32) | 0.00 (+-0.00) | 328.50 (+-7.32) | 0.70 (+-0.68) | 14.30 (+-6.06) | 11.00 (+-2.75) | 0.20 (+-0.42) |
| | **Shipment from abroad (3)** | 0.00 (+-0.00) | 7.00 (+-2.21) | 3.90 (+-2.47) | 78.30 (+-5.87) | 9.20 (+-3.65) | 1.70 (+-1.42) | 0.90 (+-0.74) |
| | **Corporate letter (4)** | 0.20 (+-0.63) | 1.80 (+-0.92) | 8.40 (+-2.37) | 3.00 (+-2.16) | 1442.30 (+-4.62) | 2.10 (+-1.79) | 3.60 (+-2.01) |
| | **Magazine (5)** | 0.10 (+-0.32) | 0.00 (+-0.00) | 9.30 (+-3.30) | 0.30 (+-0.68) | 2.00 (+-1.25) | 163.20 (+-4.34) | 1.50 (+-1.51) |
| | **Other (6)** | 0.30 (+-0.48) | 0.90 (+-0.99) | 0.50 (+-0.71) | 0.50 (+-0.71) | 11.30 (+-4.11) | 2.80 (+-1.81) | 45.30 (+-4.72) |