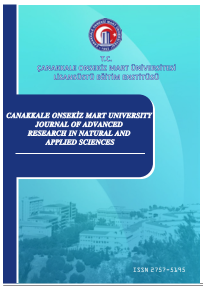
e-ISSN 2757-5195


T.C.
ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
CANAKKALE ONSEKİZ MART UNIVERSITY
JOURNAL OF ADVANCED
RESEARCH IN NATURAL AND
APPLIED SCIENCES
ISSN 2757-5195

*Çanakkale Onsekiz Mart University*
*Journal of Advanced Research in Natural and Applied Sciences*
*Open Access*

doi.org/10.28979/jarnas.953634 2022, Vol 8, Issue 1, Pages: 8-25 dergipark.org.tr/tr/pub/jarnas

# Classification of Invoice Images By Using Convolutional Neural Networks

Ömer Arslan [1],*, Sait Ali Uymaz[2]

[1] Kuveyt Türk Participation Bank Inc., Konya Research and Development (R&D) Center, Konya, Turkey
[2]Computer Engineering Department, Faculty of Engineering and Natural Sciences, Konya Technical University, Konya, Turkey

**Article History**

| | |
|---|---|
| *Received:* | 17.06.2021 |
| *Accepted:* | 18.10.2021 |
| *Published:* | 10.03.2022 |

***Araştırma Makalesi***

**Abstract** − Today, as the companies grow, the number of personnel working within the company and the number of supplier companies that the company works with are also increasing. In parallel with this increase, the amount of expenditure made on behalf of the company increases, and more invoices are created. Since the in-voices must be kept for legal reasons, physical invoices are transferred to the digital environment. Since large companies have large numbers of invoices, labor demand is higher in digitalizing invoices. In addition, as the number of invoices to be transferred to digital media increases, the number of possible errors during entry becomes more. This paper aims to automate the transfer of invoices to the digital environment. In this study, invoices be-longing to four different templates were used. Invoice images taken from a bank system were used for the first time in this study, and the original invoice dataset was prepared. Furthermore, two more datasets were obtained by applying preprocessing methods (Zero-Padding, Brightness Augmentation) on the original dataset. The Invoice classification system developed using Convolutional Neural Networks (CNN) architectures named LeNet-5, VGG-19, and MobileNetV2 was trained on three different data sets. Data preprocessing techniques such as correcting the curvature and aspect ratio of the invoices and image augmentation with variable brightness ratio were applied to create the data sets. The datasets created with preprocessing techniques have increased the classification success of the proposed models. With this proposed model, invoice images were automatically classified according to their templates using CNN architectures. In experimental studies, a classification success rate of 99.83% was achieved in training performed on the data set produced by the data augmentation method.

***Keywords*** − *Deep Learning, Convolutional Neural Networks, Image Classification, Invoice, MobileNetV2*

## Introduction

Millions of financial transactions are made every day, depending on the work of the companies. Financial transactions bring with them documents such as bank checks and invoices. It is crucial for these documents to be transferred to the digital environment and then processed automatically in terms of both security and time savings (Tang, Suen, De Yan, & Cheriet, 1995). Furthermore, financial documents created after bank transactions have similar templates. Therefore, it is possible to establish a system that can automatically recognize these financial documents, and document processing becomes easier and safer with the established system.

It may take a long time to process the documents and then obtain the necessary information written on documents. In their study, Casey, Ferguson, Mohiuddin, and Walach (1992 stated the cost of a company employee to transfer a document to a computer environment as $2. Large companies process thousands of financial documents per day. Since the information on the documents is transferred to the digital environment one by one, this process brings higher labor costs to the companies. Therefore, if the information on the document is automatically transferred to the digital environment, huge companies get rid of a very high labor cost.

[1] omer_arslan@kuveytturk.com.tr
[2] sauymaz@ktun.edu.tr
*Corresponding Author

Nowadays, it has become effortless to access computers with high processing power and storage capacity. Physical storage of documents is a costly and less secure method. Keeping the documents in the digital environment instead of physically storing them has become much easier and safer with today's technology. Subsequent access to documents stored in the digital environment is also much easier than physically storing.

Deep learning, one of the subfields of artificial intelligence, has been successfully applied in many different areas in recent years. CNNs are deep learning architectures inspired by the vision mechanism of living organisms (Gu et al., 2018). CNN architectures, which fall into the category of deep networks, are very popular, especially in image classification and pattern recognition, and have many applications applications (Liu, Fang, Zhao, Wang, & Zhang, 2015).

CNN architectures, which are widely used in image classification, have also been used in document classification studies. In the document classification study named DeepDocClassifier, a CNN architecture-based document classification system was presented using the ImageNet data set, containing millions of sample images. The system takes 3x227x227 size document images and consists of four convolutions and three fully connected layers. The system, which has seven trainable layers in total, was established with the inspiration of AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) architecture. The presented system is compared with a CNN-based method (Kang, Kumar, Ye, Li, & Doermann, 2014). In the classification study with only 20 images per class, 68.25% success was achieved. Based on the results obtained on the Tobacco-3428 data set, the system presented an average of 77.6% success in training and validation with 100 images per class. The previously proposed CNN-based approach (Kang et al., 2014) achieved an average success of 65.35% (Afzal et al., 2015).

CNN architectures are also used directly in invoice classification studies. For example, in a study that automatically classifies invoice images into three groups as handwriting, computer printout, and receipt, the CNN architecture named AlexNet (Krizhevsky et al., 2012) was used as a feature extractor. In the study, after feature extraction, various machine learning algorithms such as K Nearest Neighbor (KNN) (Brown, 2017), Random Forests (Breiman, 2001), Naive Bayes were used in the classification stage. In addition, different Cross-Validation approaches were also applied during the evaluation of the system. In the system, 1380 invoice images were used as data set, including 220 handwriting, 90 receipt, and 1070 computer printouts. During the study, 5-layer, 10-layer Cross Verification and 66% / 34% training/test split approaches were applied, and the studies were carried out separately for KNN, Random Forests, and Naive Bayes classifiers. The developed CNN-based system achieved the highest success with 98.4% if KNN was used as a classifier with a 10-layer cross-validation approach. With this high success rate, it became possible to use the established system in preprocessing stages in Optical Character Recognition(OCR) systems (Tarawneh, Hassanat, Chetverikov, Lendak, & Verma, 2019).

In this study, we propose a CNN-based system that enables the classification of invoice images that fall into financial documents. The data set used in the system was created with invoice images belonging to four different classes, and the images were taken from a bank database operating in Turkey. Three different CNN architectures, LeNet5 (LeCun, Bottou, Bengio, & Haffner, 1998), VGG19 (Simonyan & Zisserman, 2014) and, MobileNetV2 (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018) were used in the study. In the first stage of the study, CNN architectures were used as feature extractors, and in the second stage, invoices were classified according to their templates by using the extracted features.

## 2. Materials and Methods

### 2.1. Convolutional Neural Networks(CNN)

CNN is a deep learning approach developed by taking inspiration from the visual cortex of living organisms and is widely used. In recent years, CNN architectures, which were initially used in object recognition studies, are also used in text detection, semantic segmentation and, object tracking (Aloysius & Geetha, 2017). CNN architectures are similar to traditional artificial neural networks in that they consist of neurons that self-optimize through learning. The most significant difference between CNN architectures from classical artificial neural networks is that they recognize patterns in images (O'Shea & Nash, 2015). Even if CNN architectures are built in different structures depending on the area of use, they generally consist of convolution and pooling layers grouped into pieces. Subsequently, they include one or more fully connected layers in standard backpropagation neural networks (Rawat & Wang, 2017).

Three different CNN architectures were used in this study. LeNet-5, which was developed in 1998 mainly for reading handwritten digits, was used as a first architecture. LeNet-5 is a simple and small architecture that has been used to read bank checks and has achieved successful results. The other architecture used, VGG, is a deep network developed in 2014 and achieved successful results. The VGG architecture used small filters like 3x3 to extract features instead of the large filters like 9x9 and 11x11 used in previous architectures. Since invoice images are similar due to their structure, a more detailed examination is required. Small size filters allow a more detailed analysis of the images. The last architecture, MobileNetV2, is designed for use on mobile devices and has a smaller model size compared to other architectures. Therefore, it is essential to use small-size models in applications intended to work on mobile devices. For these reasons, these three CNN architectures with different structures were used in this study.

**The LeNet-5** architecture presented by LeCun et al. (1998) and developed to classify handwritten digits, can learn directly from raw pixels and requires little preprocessing in images. LeNet-5 architecture with multiple layers can be trained using the backpropagation algorithm (Gu et al., 2018). LeNet-5 architecture is a CNN architecture consisting of 7 layers with trainable parameters. The architecture includes two convolution layers, two pooling layers, two fully connected layers, and an output layer. In the past, LeNet-5 was used to recognize handwritten numbers on checks in many banks in the United States, and excellent recognition success was achieved (Wang & Gong, 2019). In the architecture, 32x32 has been chosen as the input image size. The general LeNet-5 architecture showing the recognition of the digit images is given in Figure 1.

**VGG-19** architecture was presented by Simonyan and Zisserman (2014). The VGG-16 and VGG-19 architectures are the main reason for the VGG (Visual Geometry Group) team to participate in the 2014 Imagenet competition. The team won first and second places in the localization and classification fields with VGG-16 and VGG-19 architectures, respectively (Carvalho, De Rezende, Alves, Balieiro, & Sovat, 2017). The VGG-19 architecture is similar to the VGG-16 architecture and differs from the VGG-16 architecture in that it has three additional layers. The extra three layers in the VGG-19 architecture can learn more detailed patterns from images for more effective training work (Reghunath, Nair, & Shah, 2019). As a result, VGG-19 architecture, which is very popular due to its advantages such as simplicity and easy applicability, has a very high performance in image classification studies. The VGG-19 architecture, like the VGG-16 architecture, takes 224 x 224 images as input (Xia et al., 2019).

VGG-19 architecture consists of 5 convolutional layer blocks followed by three fully connected layers. Convolutional layers use 3x3 filters and use 1 as the number of strides. Convolutional layers also use the value 1 as padding to ensure that each activation map maintains the exact spatial dimensions as the previous layer. In the architecture, the ReLU activation function is used after each convolution layer, and the maximum pooling process is usually performed to reduce the spatial dimensions. The scheme of VGG-19 architecture is shown in Figure 1.

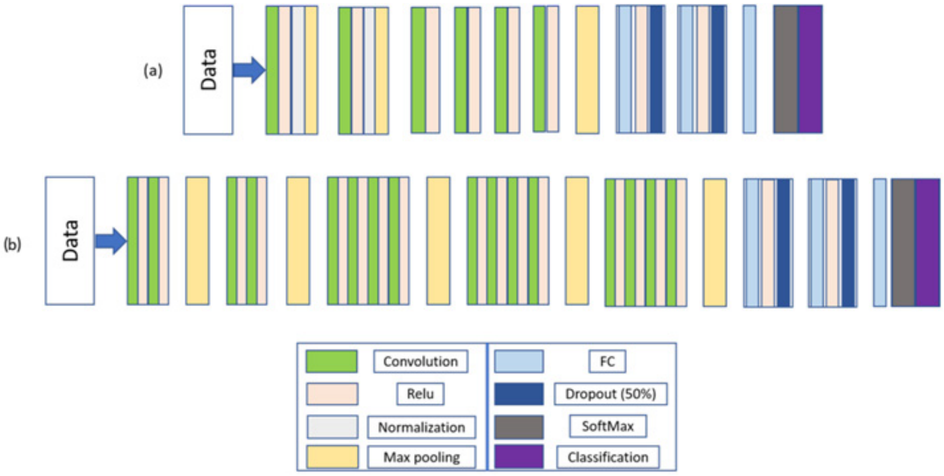


Figure 1. LeNet-5 and VGG-19 architectures (a) LeNet-5 (b) VGG-19 (Shaha & Pawar, 2018)

**MobileNetV1** architecture is a CNN architecture developed for mobile devices, reducing network cost and size. With the use of this architecture, image classification processes on mobile devices can be applied quite easily. MobileNetV2 architecture was developed on MobileNetV1 architecture in 2018. The MobileNetV2 architecture was initially designed to recognize facial features but was later trained in Google's in-house dataset (Saxen et al., 2019). Segmentation, classification, and object recognition operations can be performed using the MobileNetV2 architecture. MobileNetV2 architecture has been developed with two new features in addition to the previous architecture. The first feature is that bottlenecks can occur linearly between layers. The second feature is the development of shortcuts between bottlenecks. The general architectural scheme of the MobileNetV2 is shown in Figure 2 (Toğaçar, Cömert, & Ergen, 2021).

In deep learning algorithms, as the depth of the network increases, the gradient vanishing problem increases. The inverted residual block in the MobileNetV2 architecture uses BN (Batch Normalization) to solve the gradient problem. When ReLU is used as the activation function in architecture, the output value varies between zero and infinity. Since such an extensive range of numbers cannot be defined using low precision resolution, the ReLU6 function, an edited form of the ReLU function, is used in the MobileNetV2 (Zou, Zhao, Qin, Pan, & Li, 2020).

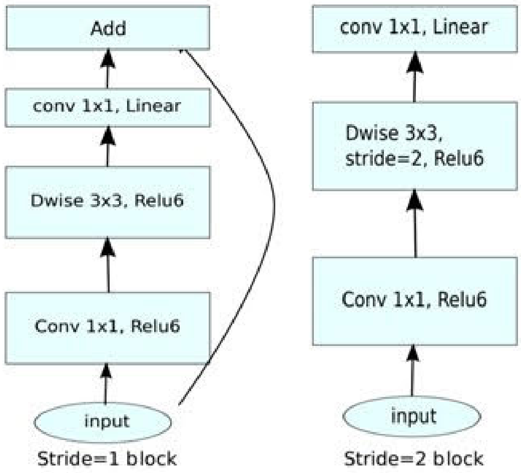


Figure 2. General architectural scheme of the MobileNetV2 (Sandler et al., 2018)

### 2.2. Data Preprocessing

The invoice images used in the training phase of the system belong to transactions such as overtime and travel performed by bank personnel. The invoices are scanned using a scanner or sent to the system without preprocessing after the photograph is taken using the mobile device camera. Therefore, the data sets used in this study were prepared by processing the invoice images that were taken from the system.

Detection of Images That Do not Contain Invoice Data

When the invoice images are scanned double-sided by the personnel using a scanner, two images are obtained. One of the images generally does not contain any data that is related to the invoice. In addition, images taken by users using a mobile device camera may sometimes have images that are not related to the invoice.

First, the typical invoice terms were determined by applying OCR to all images. Then, by applying OCR to the images for the second time, images containing no text or no invoice terms were determined and removed from the data set. During the application of OCR to the images, an open-source OCR system called Tesseract was used. Tesseract was developed by Hewlett Packard in the 1990s and became open source in late 2005 (Sidhwa, Kulshrestha, Malhotra, & Virmani, 2018). Sample images automatically removed from the data set after OCR application are shown in Figure 3.

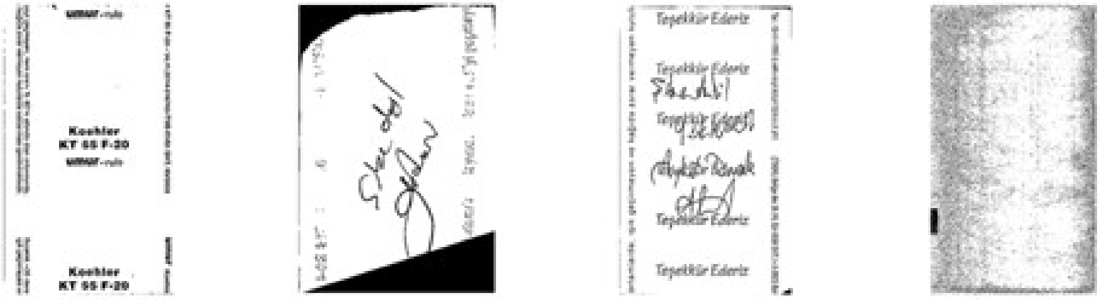


Figure 3. Sample images that were removed from the dataset as they do not contain meaningful data after OCR

### 2.2.2. Correcting Of Oblique Angle Images

The images in the data set are taken from different angles by different users. Oblique images affect both OCR performance and feature extraction performance made by CNN. To prevent these possible problems, the angle of the texts in the images was calculated, and the tilt of the curved images was corrected. During the tilt correction study, firstly, the images in the data set were converted to the black and white format. Then, tilt calculation was made using the edge detection operator named Canny and Hough transform.

#### 2.2.2.1. Canny Edge Detection Operator

The Canny operator was developed by John F. and is used to detect edges in images with its algorithm consisting of multiple steps (Khan & Mufti, 2016). The Canny algorithm consists of four steps. In the first step, noise reduction is performed using the Gaussian filter. In the second step, the gradient is calculated, and in the third step, the pixels that are not edge candidates and are not the maximum pixels are eliminated. Finally, the last step decides whether the pixels are edge pixels or not (Ha & Shakeri, 2016). After the invoice images were converted to black and white format, the edge detection process was carried out by applying the Canny operator.

#### 2.2.2.2. Hough Transform

Hough transform is a feature extraction technique used in digital image processing studies. Although Hough transform is used to detect lines in images, it can also be used to detect the locations of random shapes in the image. The Hough transform is designed to detect lines and uses the parametric representation of the line as shown in Equation (2.1).

$$\rho=x\cos\theta+y\sin\theta \tag{2.1}$$

The  in the formula expresses the distance from the vector perpendicular to the line.  denotes the angle between this vector and the x-axis (Chunhavittayatera, Chitsobhuk, & Tongprasert, 2006). The representation of the Hough space belonging to the line is shown in Figure 4.

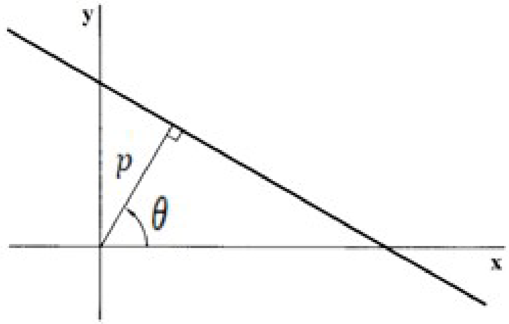


Figure 4. Representation of the Hough space of the line (Duda & Hart, 1972)

Invoice images were converted to black and white format, and edge detection was performed by applying the Canny operator. Later, by applying the Hough transform, lines were drawn to the rows in the image, and the tilt of each row was calculated and collected. After the correct tilts were collected, the image inclination angle was calculated. Then the image was rotated by using a computed tilt angle. Conversion of a sample invoice image in the data set to black and white format, Canny operator application, and Hough transformation implementation are shown in Figure 5.

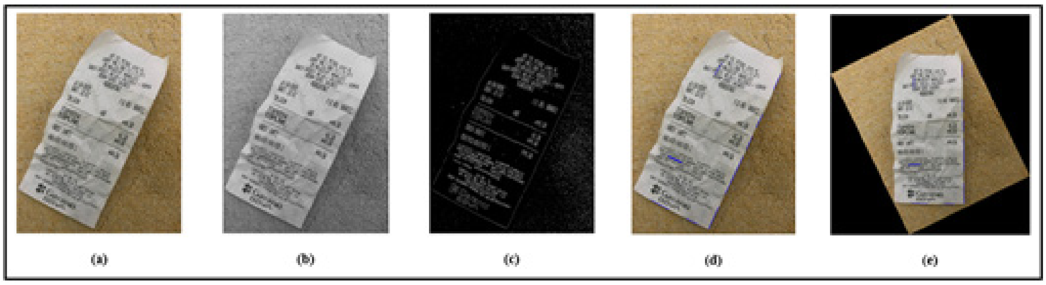


Figure 5. Steps of applying the Canny Operator and Hough Transform to a sample invoice image. (a) Original image. (b) Black and white image. (c) Image with Canny operator applied. (d) The image containing lines that drawn using Hough transform. (e) Corrected image by using the tilt angle obtained by taking the average tilt of the lines.

### 2.2.3. Zero Padding Application

Since CNN architectures take fixed-size images as input, the images in the training set need to be resized. However, while resizing images with very different aspect ratios, there is a loss of detail problem in some images. To solve the aspect ratio problem, the images were resized to be equal in width and height using the zero-padding method. According to the zero-padding process, the image is resized by placing fixed-value pixels on the edges of the image. With this method, the spatial distribution in the image edges is changed changed (Nguyen et al., 2019). The zero-padding process has two significant advantages. The first advantage; if images resized using zero-padding are then resized again, the aspect ratio will be preserved, so there will be no loss of features or deformation. Another advantage is that it speeds up calculation processes and reduces calculation costs.

### 2.2.4. Data Augmentation

Raw invoice images belong to different users, and each was taken by different mobile cameras or transferred to digital media using a scanner. Therefore, the amount of light in each image is different. Therefore, the proposed invoice classification system should be capable of recognizing the invoices with varying values of light. So that, the system was trained on images that contain different amounts of light. For this purpose, new images were obtained by changing the brightness values of the images in the dataset obtained by applying zero-padding. With this image augmentation technique, the data set size was increased from 4000 to 12000 images. The image enhancement process was carried out by using the ImageDataGenerator class in the library named Keras, and [0.2, 1] was used as the brightness range.

### 2.3. Datasets

Invoice images taken from a bank system were used for the first time in this study, and the invoice dataset was prepared. While preparing the dataset, some corrections (Hough Transform) were made on the old invoice images in the system. This process has been applied to a small number of images in the bank system. The bank system uploads new images by applying these processes. Furthermore, two more datasets were obtained by applying preprocessing methods (Zero-Padding, Brightness Augmentation) on the original dataset.

In the data set created from invoice images, 4000 invoice images belong to four classes. Class names of invoice images and invoice numbers for each class are given in Table 1.

Table 1

Invoice classes and number of invoice images per class

| Invoice Class | Invoice Image Count |
|---|---|
| CardFilling | 1000 |
| Parking | 1000 |
| Taxi | 1000 |
| Meal | 1000 |
| Total | 4000 |

Three different data sets were used in the project. The original data set was obtained by using images taken from a private bank's database. The correction process is applied while the images are saved in the bank database. After the images were retrieved from the bank database, the Canny and Hough Transform processes were applied on only a few old invoice images. Therefore, the images in the original dataset consist of raw images taken from the bank. Moreover, Canny and Hough operations, which were used only while generating the data set, are not used again during the operation of the system. Therefore, these processes do not bring additional costs to the devices on which the application will run.

By applying the Zero-Padding process to the images in the original dataset, the image aspect ratios were equalized, and thus, the BAFGV2 dataset was obtained. BAFGV2 dataset was produced to avoid the loss of detail problem when resizing images in different aspect ratios.

BAFGV3, the last data set used in the project, was obtained by augmenting the images of the BAFGV2 data set based on the brightness values. Since real-world invoice images will have different brightness values, the system needs to be prepared for this. Therefore, to make the system work stably on images of varying brightness, the BAFGV3 data set was created.

### 2.3.1. Original Dataset

The Original dataset includes invoice images taken from a private bank's database. An example invoice image of each class in the Original dataset is given in Figure 6.

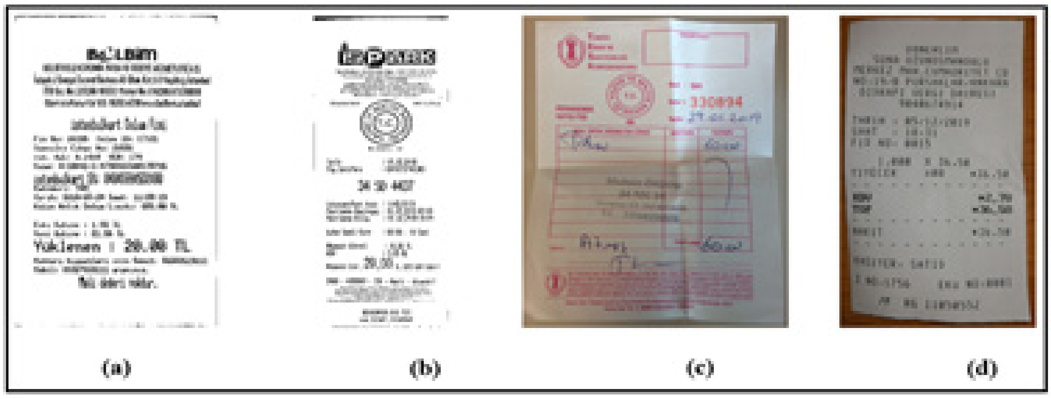


Figure 6. Image samples of each class in the Original dataset. (a) CardFilling receipt. (b) Parking receipt. (c) Taxi receipt. (d) Meal receipt.

### 2.3.2. BAFGV2 Dataset

The BAFGV2 data set was created by applying the zero-padding process to the images in the Original data set. The BAFGV2 data set was obtained by adding blue pixels around the images of Original to make them square. Therefore, each image in the BAFGV2 data set is square-shaped, and the aspect ratio is equal. An example of an invoice for each class in the BAFGV2 data set is shown in Figure 7.

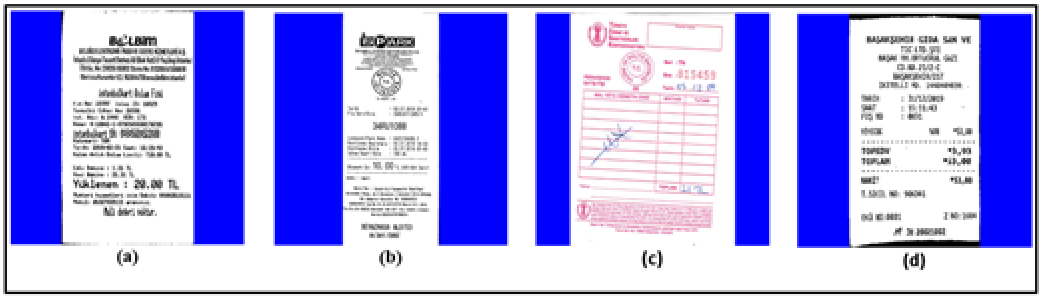


Figure 7. Sample invoice images of the BAFGV2 dataset. (a) CardFilling receipt. (b) Parking receipt. (c) Taxi receipt. (d) Meal receipt

### 2.3.3. BAFGV3 Dataset

By using the brightness-based data augmentation method, two more images with different brightness values were obtained from each image in the BAFGV2 data set. Therefore, with this image augmentation technique, the data set size from 4000 was increased to 12000 images.

An example process for obtaining two different images from each image by changing the brightness value is shown in Figure 8.

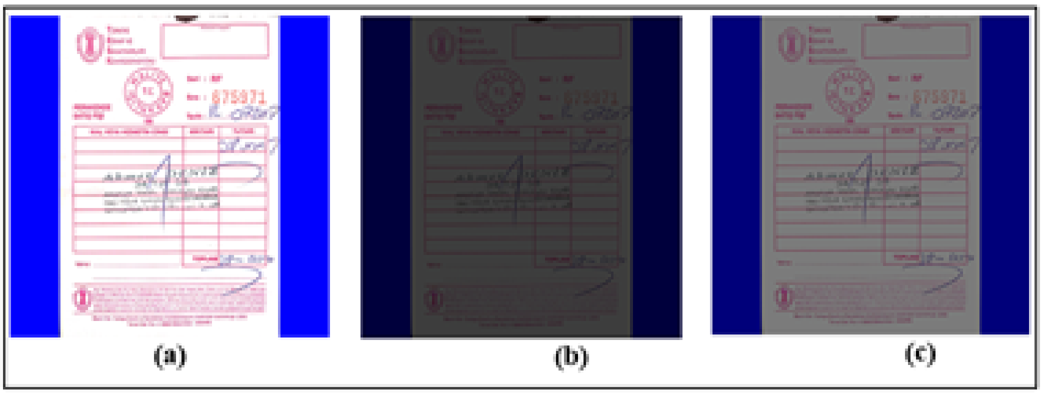


Figure 8. Sample images that were obtained with brightness-based image augmentation. (a) Original image (b) First image obtained (c) Second image obtained

## 3. Results and Discussion

The proposed study performed feature extraction and classification on three data sets using LeNet-5, VGG-19, and MobileNetV2 CNN architectures. In the studies, the invoice images taken from the database of a bank operating in Turkey and containing Turkish data were used as the data set. 5-Fold cross-validation technique was used in the training studies conducted on three different data sets using three different CNN architectures, and a total of 45 training studies were carried out.

### 3.1. Experimental Studies on the Original Dataset

A total of 15 training studies were carried out on the Original data set. Success rates and success charts of the training are presented in the rest of the section. The training and validation success rates stated in the graphs represent the success rates achieved in the last training epoch.

The LeNet-5 architecture was used as a feature extractor on the Original data set using the 5-Fold cross-validation technique. The results of the five training studies are presented in Table 2. The confusion matrix of predictions made in the 1st layer during the test is shown in Table 3.

Table 2

Success rates of studies that are using LeNet-5 architecture

| | Training Accuracy | Validation Accuracy | Test Accuracy | Training Time | Model Size |
|---|---|---|---|---|---|
| Fold - 1 | %94.22 | %91.41 | %93.500 | 12 minutes | 22 MB |
| Fold - 2 | %82.27 | %86.09 | %71.125 | 12 minutes | 22 MB |
| Fold - 3 | %93.24 | %93.44 | %90.625 | 12 minutes | 22 MB |
| Fold - 4 | %87.93 | %93.12 | %85.375 | 12 minutes | 22 MB |
| Fold - 5 | %90.90 | %86.87 | %78.875 | 12 minutes | 22 MB |
| Average | %89.712 | %90.186 | %83.900 | 12 minutes | 22 MB |

Table 3

The confusion matrix of predictions made in the most successful layer using Lenet-5 architecture

| | | Actual Values | | | |
|---|---|---|---|---|---|
| | | CardFilling | Parking | Taxi | Meal |
| Predicted Values | CardFilling | 198 | 0 | 1 | 5 |
| | Parking | 0 | 196 | 0 | 0 |
| | Taxi | 0 | 0 | 165 | 6 |
| | Meal | 2 | 4 | 34 | 189 |

As seen in Table 2, the average validation accuracy of the LeNet-5 model on the Original dataset is 90%, and the test accuracy is 83.9%. According to Table 3, LeNet-5 achieved more success in detecting CardFilling and Parking classes than other classes.

As a second architecture, VGG-19 architecture was used as a feature extractor on the Original data set by using the 5-Fold cross-validation technique. The results of the five training studies are presented in Table 4. In addition, the confusion matrix of predictions made in the 5th layer during the test is shown in Table 5.

Table 4

Success rates of studies that are using VGG-19 architecture

| | Training Accuracy | Validation Accuracy | Test Accuracy | Training Time | Model Size |
|---|---|---|---|---|---|
| Fold - 1 | %24.61 | %25.00 | %25.00 | 192 minutes | 545 MB |
| Fold - 2 | %25.16 | %43.44 | %25.00 | 192 minutes | 545 MB |
| Fold - 3 | %24.61 | %25.00 | %43.37 | 192 minutes | 545 MB |
| Fold - 4 | %24.57 | %25.00 | %38.25 | 192 minutes | 545 MB |
| Fold - 5 | %23.83 | %25.00 | %62.87 | 192 minutes | 545 MB |
| Average | %24.55 | %28.68 | %38.90 | 192 minutes | 545 MB |

Table 5

Confusion matrix of predictions made in the most successful layer using VGG-19 architecture

| | | Actual Values | | | |
|---|---|---|---|---|---|
| | | CardFilling | Parking | Taxi | Meal |
| Predicted Values | CardFilling | 100 | 0 | 6 | 13 |
| | Parking | 97 | 199 | 17 | 102 |
| | Taxi | 2 | 1 | 177 | 58 |
| | Meal | 1 | 0 | 0 | 27 |

According to Table 4, the highest test accuracy of the VGG-19 architecture was achieved in Fold-5. However, the average test accuracy is 38.9%. Thus, the experimental results of the VGG-19 model on the Original dataset are lower than that of LeNet-5.

As a third architecture, MobileNetV2 architecture was used as a feature extractor on the Original data set using the 5-Fold cross-validation technique. The results of the five training studies are presented in Table 6. The confusion matrix of predictions made in the 5th layer during the test is shown in Table 7.

Table 6

Success rates of studies that are using MobileNetV2 architecture

| | Training Accuracy | Validation Accuracy | Test Accuracy | Training Time | Model Size |
|---|---|---|---|---|---|
| Fold - 1 | %99.65 | %95.54 | %41.87 | 58 minutes | 18 MB |
| Fold - 2 | %99.96 | %95.63 | %47.12 | 58 minutes | 18 MB |
| Fold - 3 | %98.96 | %99.22 | %25.25 | 58 minutes | 18 MB |
| Fold - 4 | %99.96 | %83.28 | %41.75 | 58 minutes | 18 MB |
| Fold - 5 | %98.70 | %88.45 | %53.87 | 58 minutes | 18 MB |
| Average | %99.44 | %92.42 | %41.97 | 58 minutes | 18 MB |

Table 7

The confusion matrix of predictions made in the most successful layer using MobileNetV2 architecture

| | | Actual Values | | | |
|---|---|---|---|---|---|
| | | CardFilling | Parking | Taxi | Meal |
| Predicted Values | CardFilling | 44 | 1 | 24 | 162 |
| | Parking | 152 | 198 | 3 | 16 |
| | Taxi | 3 | 1 | 173 | 6 |
| | Meal | 1 | 0 | 0 | 16 |

As shown in Table 6, the highest test accuracy was obtained in Fold-5. Although 53.87% test success is achieved in this layer, validation and training accuracies are much higher. This success difference is because the Original dataset contains images that have variable aspect ratios. Furthermore, these aspect ratios cause the loss of detail when resizing images before the CNN network. Since MobileNetV2 is much deeper than the other two architectures, underachievement due to the loss of detail is more in the MobileNetV2 architecture. The confusion matrix of the studies carried out in Fold-5 is shown in Table 7. Accordingly, the highest accuracy value was obtained in Parking images.

LeNet-5 CNN architecture achieved the highest accuracy rates in studies with the Original dataset. LeNet-5 architecture achieved 93.50% test accuracy in this dataset. Images with very different aspect ratios are used in the Original dataset. While the images were sent as input to the CNN network, they were resized to a standard size of 224x224, so some images lost detail, and the distinctiveness of the invoice images decreased. For this reason, the deeper networks VGG-19 and MobileNetV2 architectures showed low test accuracy because of the overfitting problem.

### 3.2. Experimental Studies on the BAFGV2 Dataset

By adding blue pixels to the images in the Original dataset, their aspect ratios were equalized, and the Original dataset was obtained. The aim of the studies carried out in this section is to observe the success rates of the training studies to be done with the images that have the same aspect ratio instead of the irregular aspect ratio. Moreover, in this section, the sizes of the models formed after the training studies are also compared. This comparison gives an idea about whether the resulting model is suitable for use in mobile devices.

In the first studies, LeNet-5 architecture was used as a feature extractor on the BAFGV2 dataset by using the 5-Fold cross-validation technique. The results of the five training studies are presented in Table 8. The confusion matrix of predictions made in the 4th layer during the test is shown in Table 9.

Table 8

Success rates of studies that are using LeNet-5 architecture

| | Training Accuracy | Validation Accuracy | Test Accuracy | Training Time | Model Size |
|---|---|---|---|---|---|
| Fold - 1 | %92.51 | %93.59 | %89.226 | 14 minutes | 22 MB |
| Fold - 2 | %74.79 | %79.11 | %74.651 | 14 minutes | 22 MB |
| Fold - 3 | %78.24 | %78.12 | %75.285 | 14 minutes | 22 MB |
| Fold - 4 | %92.67 | %95.89 | %93.662 | 14 minutes | 22 MB |
| Fold - 5 | %93.24 | %92.60 | %92.522 | 14 minutes | 22 MB |
| Average | %86.290 | %87.862 | %85.069 | 14 minutes | 22 MB |

Table 9

The confusion matrix of predictions made in the most successful layer using LeNet-5 architecture

| | | Actual Values | | | |
|---|---|---|---|---|---|
| | | CardFilling | Parking | Taxi | Meal |
| Predicted Values | CardFilling | 188 | 0 | 0 | 6 |
| | Parking | 0 | 198 | 0 | 3 |
| | Taxi | 1 | 0 | 188 | 19 |
| | Meal | 11 | 2 | 8 | 165 |

According to Table 8, the highest test accuracy was obtained in Fold 4. The training and validation accuracy values in this layer are also similar. Therefore, the training and test success difference problem did not occur in this section. This enhancement is because the BAFGV2 dataset consists of images with equal aspect ratios. In this way, no detail was lost during the resizing stage before the CNN network, and there was no problem of low success.

As a second architecture, VGG-19 architecture was used as a feature extractor on the BAFGV2 data set by using the 5-Fold cross-validation technique. The results of the five training studies are presented in Table 10. In addition, the confusion matrix of predictions made in the 5th layer during the test is shown in Table 11.

Table 10

Success rates of studies that are using VGG-19 architecture

| | Training Accuracy | Validation Accuracy | Test Accuracy | Training Time | Model Size |
|---|---|---|---|---|---|
| Fold - 1 | %95.51 | %96.22 | %97.71 | 187 minutes | 545 MB |
| Fold - 2 | %98.36 | %96.55 | %96.07 | 187 minutes | 545 MB |
| Fold - 3 | %87.33 | %98.19 | %96.83 | 187 minutes | 545 MB |
| Fold - 4 | %75.95 | %80.10 | %74.77 | 187 minutes | 545 MB |
| Fold - 5 | %97.23 | %98.19 | %97.71 | 187 minutes | 545 MB |
| Average | **%90.87** | **%93.85** | **%92.62** | 187 minutes | 545 MB |

Table 11

The confusion matrix of predictions made in the most successful layer using VGG-19 architecture

| | | Actual Values | | | |
|---|---|---|---|---|---|
| | | CardFilling | Parking | Taxi | Meal |
| Predicted Values | CardFilling | 198 | 0 | 0 | 1 |
| | Parking | 0 | 199 | 0 | 3 |
| | Taxi | 0 | 0 | 190 | 5 |
| | Meal | 2 | 1 | 6 | 184 |

According to Table 10, the average test accuracy was 92.62% in studies performed on BAFGV2 using VGG-19. This success rate was 85.06% in LeNet-5. The training of the VGG-19 network took 187 minutes, while the training of the LeNet-5 network took only 14 minutes. When Table 11 is examined, the results support the values given in Table 10.

As a third architecture, MobileNetV2 architecture was used as a feature extractor on the BAFGV2 data set using the 5-Fold cross-validation technique. The results of the five training studies are presented in Table 12. In addition, the confusion matrix of predictions made in the 3rd layer during the test is shown in Table 13.

Table 12

Success rates of studies that are using MobileNetV2 architecture

| | Training Accuracy | Validation Accuracy | Test Accuracy | Training Time | Model Size |
|---|---|---|---|---|---|
| Fold - 1 | %99.88 | %98.19 | %62.61 | 51 minutes | 18 MB |
| Fold - 2 | %100.0 | %99.34 | %61.21 | 51 minutes | 18 MB |
| Fold - 3 | %99.76 | %95.56 | %79.46 | 51 minutes | 18 MB |
| Fold - 4 | %99.24 | %97.11 | %67.42 | 51 minutes | 18 MB |
| Fold - 5 | %99.84 | %76.15 | %75.79 | 51 minutes | 18 MB |
| Average | %99.88 | %98.19 | %62.61 | 51 minutes | 18 MB |

Table 13

The confusion matrix of predictions made in the most successful layer using MobileNetV2 architecture

| | | Actual Values | | | |
|---|---|---|---|---|---|
| | | CardFilling | Parking | Taxi | Meal |
| Predicted Values | CardFilling | 189 | 0 | 0 | 2 |
| | Parking | 0 | 183 | 0 | 0 |
| | Taxi | 0 | 0 | 64 | 0 |
| | Meal | 11 | 17 | 132 | 191 |

It can be seen in Table 12 that the highest test accuracy was obtained in Fold-3. However, the training and validation accuracy values in this layer are much higher. This difference usually indicates that an overfitting problem has occurred. Since the BAFGV2 dataset contains images with equal aspect ratios, there is no loss of detail during resizing. However, since the MobileNetV2 architecture is much deeper than the other two networks and the invoice images are similar, the overFitting problem occurred, and a lower test success was achieved than the training success.

In the studies carried out on the BAFGV2 data set, VGG-19 achieved the highest test success with 97.71%. The BAFGV2 dataset contains images with an aspect ratio of 1. The zero-Padding application was used in arranging the aspect ratio as 1. Since the images in this dataset have a fixed aspect ratio, there was less loss of detail when resizing at the CNN input, and much higher test success was achieved.

### 3.3. Experimental Studies on the BAFGV3 Dataset

The BAFGV3 dataset was obtained by augmenting the images in the BAFGV2 dataset based on the brightness values, and it consists of 12000 images. Processing and classifying images taken in environments with varying amounts of light is a challenging process. Therefore, training studies were also carried out with the BAFGV3 dataset to ensure that the proposed invoice classification system works more stable against images containing variable light amounts.

The LeNet-5 architecture was used as a feature extractor in the studies performed on the BAFGV3 data set, and the 5-Fold cross-validation technique was used. The results of the five training studies are presented in Table 14. In addition, the confusion matrix of predictions made in the 1st layer during the test is shown in Table 15.

Table 14

Success rates of studies that are using LeNet-5 architecture

| | Training Accuracy | Validation Accuracy | Test Accuracy | Training Time | Model Size |
|---|---|---|---|---|---|
| Fold - 1 | %96.44 | %98.73 | %98.944 | 42 minutes | 22 MB |
| Fold - 2 | %97.40 | %98.68 | %98.775 | 42 minutes | 22 MB |
| Fold - 3 | %98.18 | %98.41 | %98.479 | 42 minutes | 22 MB |
| Fold - 4 | %98.13 | %98.41 | %98.564 | 42 minutes | 22 MB |
| Fold - 5 | %98.77 | %98.52 | %98.437 | 42 minutes | 22 MB |
| Average | %97.78 | %98.550 | %98.640 | 42 minutes | 22 MB |

Table 15

The confusion matrix of predictions made in the most successful layer using LeNet-5 architecture

| | | Actual Values | | | |
|---|---|---|---|---|---|
| | | CardFilling | Parking | Taxi | Meal |
| Predicted Values | CardFilling | 599 | 0 | 1 | 3 |
| | Parking | 0 | 599 | 0 | 0 |
| | Taxi | 0 | 0 | 573 | 5 |
| | Meal | 1 | 1 | 14 | 572 |

Table 14 shows that the training time of the LeNet-5 network on the BAFGV3 dataset is 42 minutes and the model size is 22 MB. Training, validation, and test achievements are similar. An average of 98.64% test success was achieved in the studies, and this success was the highest average success achieved with LeNet-5. Since the BAFGV3 dataset includes images augmented by varying amounts of light, and real-world examples include images with different amounts of light, higher success has been achieved on this dataset.

As a second architecture, VGG-19 architecture was used as a feature extractor on the BAFGV3 data set by using the 5-Fold cross-validation technique. The results of the five training studies are presented in Table 16. Moreover, the confusion matrix of predictions made in the 4th layer during the test is shown in Table 17.

Table 16

Success rates of studies that are using VGG-19 architecture

| | Training Accuracy | Validation Accuracy | Test Accuracy | Training Time | Model Size |
|---|---|---|---|---|---|
| Fold - 1 | %99.79 | %99.63 | %99.61 | 560 minutes | 545 MB |
| Fold - 2 | %99.81 | %96.40 | %97.00 | 560 minutes | 545 MB |
| Fold - 3 | %99.52 | %99.47 | %99.49 | 560 minutes | 545 MB |
| Fold - 4 | %99.77 | %99.84 | %99.83 | 560 minutes | 545 MB |
| Fold - 5 | %99.43 | %99.52 | %99.61 | 560 minutes | 545 MB |
| Average | %99.66 | %98.97 | %99.11 | 560 minutes | 545 MB |

Table 17

The confusion matrix of predictions made in the most successful layer using VGG-19 architecture

| | | Actual Values | | | |
|---|---|---|---|---|---|
| | | CardFilling | Parking | Taxi | Meal |
| Predicted Values | CardFilling | 600 | 0 | 1 | 0 |
| | Parking | 0 | 600 | 0 | 2 |
| | Taxi | 0 | 0 | 586 | 1 |
| | Meal | 0 | 0 | 0 | 577 |

According to Table 16, an average test success rate of 99.11% was obtained. In addition, it can be seen in Fold-4 values that 99.83% test success was achieved. This value was the highest test success among all studies. Therefore, the VGG-19 architecture has been the most successful network. However, training of the VGG-19 network on the BAFGV3 dataset took 560 minutes, and the model size is 545 MB. Therefore, it would not be advantageous to choose the VGG-19 network where model size and training time are essential metrics.

As a third architecture, MobileNetV2 architecture was used as a feature extractor on the BAFGV3 data set using the 5-Fold cross-validation technique. The results of the five training studies are presented in Table 18. In addition, the confusion matrix of predictions made in the 4th layer during the test is presented in Table 19.

Table 18

Success rates of studies that are using MobileNetV2 architecture

| | Training Accuracy | Validation Accuracy | Test Accuracy | Training Time | Model Size |
|---|---|---|---|---|---|
| Fold - 1 | %99.97 | %100.0 | %80.52 | 189 minutes | 18 MB |
| Fold - 2 | %99.99 | %99.79 | %82.63 | 189 minutes | 18 MB |
| Fold - 3 | %99.97 | %99.95 | %77.82 | 189 minutes | 18 MB |
| Fold - 4 | %99.96 | %100.0 | %95.98 | 189 minutes | 18 MB |
| Fold - 5 | %99.96 | %99.79 | %79.29 | 189 minutes | 18 MB |
| Average | %99.97 | %99.90 | %83.25 | 189 minutes | 18 MB |

Table 19

The confusion matrix of predictions made in the most successful layer using MobileNetV2 architecture

| | | Actual Values | | | |
|---|---|---|---|---|---|
| | | CardFilling | Parking | Taxi | Meal |
| Predicted Values | CardFilling | 584 | 13 | 15 | 37 |
| | Parking | 0 | 578 | 0 | 0 |
| | Taxi | 12 | 0 | 568 | 1 |
| | Meal | 4 | 9 | 4 | 542 |

According to Table 18, MobileNetV2 achieved the highest average test success of 83.25% in the BAFGV3 dataset. This success rate shows that the difference in success between training and test studies on the BAFGV3 dataset has decreased.

The BAFGV3 dataset was obtained by randomly changing the brightness values of the images in the BAFGV2 dataset. As a result, the BAFGV3 dataset is three times greater than the size of the other two datasets. In the studies carried out, VGG-19 achieved the highest success with 99.83%. In addition, this success achieved by VGG-19 is the highest success achieved in all studies within the proposed system.

In the studies, the best success rates were obtained in the BAFGV3 data set. The best results are shown in Table 20 according to the CNN architectures used.

Table 20

Best success rates achieved

| | Training Accuracy | Validation Accuracy | Test Accuracy |
|---|---|---|---|
| VGG-19 | %99.77 | %99.84 | %99.83 |
| MobileNetV2 | %99.96 | %100.0 | %95.98 |
| LeNet-5 | %96.44 | %98.73 | %98.94 |

The model dimensions obtained in the studies were examined within the scope of the study. As the model size, the value in MB of the space occupied in the memory is given. Model size does not change according to the size of the data set, but varies according to the number of model parameters and the model working structure. The model sizes and other training parameters are given in Table 21.

Table 21

Model sizes and other training parameters

| | Model Size | Number of Parameters | Epoch | Batch Size | Learning Rate | Optimizer | Loss Function |
|---|---|---|---|---|---|---|---|
| VGG-19 | 545 MB | 143,667,240 | 10 | 32 | 0.01 | SGD | Categorical Cross Entropy |
| LeNet-5 | 22 MB | 60,815 | 10 | 32 | 0.01 | SGD | Categorical Cross Entropy |
| MobileNetV2 | 18 MB | 3,538,984 | 10 | 32 | 0.01 | SGD | Categorical Cross Entropy |

The data set used in the present study was created using invoice images taken from a bank database and was used for the first time in this study. Since a known data set in the literature is not used, the study is an original study. However, there are some similar studies presented on the scope of invoice classification. For example, in a study that automatically classifies invoice images into three groups as handwritten, computer printout, and receipt, a CNN architecture named AlexNet was used as a feature extractor. In this CNN-based system, the highest success rate of 98.4% was achieved using KNN as the classifier with the 10-Fold cross-validation approach.

## 4. Conclusion

In the study, a CNN-based system has been developed that classifies invoice images containing Turkish data according to their templates. In the study, LeNet-5, VGG-19, and MobileNetV2 architectures were trained using three separate data sets. Since the 5-Fold cross-validation technique was used during the study, 45 training studies were carried out. Then, the most successful layers were selected based on the data set and CNN architecture. Testing and validation success rates were taken into consideration while comparing the layer successes.

The training studies obtained the lowest success rates were obtained on the Original dataset, which contains images with different aspect ratios. The reason for this low success rate is the loss of detail on the images during the resizing phase. The BAFGV2 dataset, which includes images with equal aspect ratios, achieved higher successes because there is no loss of detail problem. The highest accuracy rates were obtained in the BAFGV3 dataset, which was acquired by increasing the BAFGV2 dataset using variable brightness ratios. The 99.83% success achieved by using the VGG-19 architecture in the BAFGV3 dataset is the highest test success achieved throughout all studies.

The CNN architectures used in the studies were also compared in terms of resource efficiency. The model size must be small, especially in models intended for use in mobile devices. Considering the model dimensions obtained in the training studies, the model produced by the MobileNetV2 architecture was the model with the smallest model size with 18 MB, while the model size produced by the VGG-19 was 545 MB.

### Acknowledgement

The author received no specific funding for this study.

### Author Contributions

Ömer ARSLAN: Graduated M.Sc. student. Obtained the data, performed the experimental studies, and wrote the paper.

Sait Ali UYMAZ: Thesis supervisor. Conceived and designed the analysis, reviewed and wrote the paper.

### Conflicts of Interest

The authors declare no conflict of interest.

### References

Afzal, M. Z., Capobianco, S., Malik, M. I., Marinai, S., Breuel, T. M., Dengel, A., & Liwicki, M. (2015). Deepdocclassifier: Document classification with deep convolutional neural network. Paper presented at the 2015 13th international conference on document analysis and recognition (ICDAR). DOI: https://doi.org/10.1109/ICDAR.2015.7333933

Aloysius, N., & Geetha, M. (2017). A review on deep convolutional neural networks. Paper presented at the 2017 International Conference on Communication and Signal Processing (ICCSP). DOI: https://doi.org/10.1109/ICCSP.2017.8286426

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32. DOI: https://doi.org/10.1023/A:1010933404324

Brown, J. M. (2017). Predicting math test scores using k-nearest neighbor. Paper presented at the 2017 IEEE Integrated STEM Education Conference (ISEC). DOI: https://doi.org/10.1109/ISECon.2017.7910221

Carvalho, T., De Rezende, E. R., Alves, M. T., Balieiro, F. K., & Sovat, R. B. (2017). Exposing computer generated images by eye's region classification via transfer learning of VGG19 CNN. Paper presented at the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). DOI: https://doi.org/10.1109/ICMLA.2017.00-47

Casey, R., Ferguson, D., Mohiuddin, K., & Walach, E. (1992). Intelligent forms processing system. Machine Vision and Applications, 5(3), 143-155. DOI: https://doi.org/10.1007/BF02626994

Chunhavittayatera, S., Chitsobhuk, O., & Tongprasert, K. (2006). Image registration using Hough transform and phase correlation. Paper presented at the 2006 8th International Conference Advanced Communication Technology. DOI: https://doi.org/10.1109/ICACT.2006.206134

Duda, R. O., & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM, 15(1), 11-15. DOI: https://doi.org/10.1145/361237.361242

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Cai, J. (2018). Recent advances in convolutional neural networks. Pattern Recognition, 77, 354-377. DOI: https://doi.org/10.1016/j.patcog.2017.10.013

Ha, P. S., & Shakeri, M. (2016). License Plate Automatic Recognition based on edge detection. Paper presented at the 2016 Artificial Intelligence and Robotics (IRANOPEN). DOI: https://doi.org/10.1109/RIOS.2016.7529509

Kang, L., Kumar, J., Ye, P., Li, Y., & Doermann, D. (2014). Convolutional neural networks for document image classification. Paper presented at the 2014 22nd International Conference on Pattern Recognition. DOI: https://doi.org/10.1109/ICPR.2014.546

Khan, M., & Mufti, N. (2016). Comparison of various edge detection filters for ANPR. Paper presented at the 2016 Sixth International Conference on Innovative Computing Technology (INTECH). DOI: https://doi.org/10.1109/INTECH.2016.7845061

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Paper presented at the Advances in neural information processing systems. DOI: https://doi.org/10.1145/3065386

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324. DOI: https://doi.org/10.1109/5.726791

Liu, T., Fang, S., Zhao, Y., Wang, P., & Zhang, J. (2015). Implementation of training convolutional neural networks. Retrieved from: https://arxiv.org/abs/1506.01195

Nguyen, A.-D., Choi, S., Kim, W., Ahn, S., Kim, J., & Lee, S. (2019). Distribution Padding in Convolutional Neural Networks. Paper presented at the 2019 IEEE International Conference on Image Processing (ICIP). DOI: https://doi.org/10.1109/ICIP.2019.8803537

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. Retrieved from: https://arxiv.org/abs/1511.08458

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. Neural computation, 29(9), 2352-2449. DOI: https://doi.org/10.1162/NECO_a_00990

Reghunath, A., Nair, S. V., & Shah, J. (2019). Deep learning based Customized Model for Features Extraction. Paper presented at the 2019 International Conference on Communication and Electronics Systems (ICCES). DOI: https://doi.org/10.1109/ICCES45898.2019.9002299

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition. DOI: https://doi.org/10.1109/CVPR.2018.00474

Saxen, F., Werner, P., Handrich, S., Othman, E., Dinges, L., & Al-Hamadi, A. (2019). Face attribute detection with mobilenetv2 and nasnet-mobile. Paper presented at the 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA). DOI: https://doi.org/10.1109/ISPA.2019.8868585

Shaha, M., & Pawar, M. (2018). Transfer learning for image classification. Paper presented at the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA). DOI: https://doi.org/10.1109/ICECA.2018.8474802

Sidhwa, H., Kulshrestha, S., Malhotra, S., & Virmani, S. (2018). Text extraction from bills and invoices. Paper presented at the 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). DOI: https://doi.org/10.1109/ICACCCN.2018.8748309

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. Retrieved from: https://arxiv.org/abs/1409.1556

Tang, Y. Y., Suen, C. Y., De Yan, C., & Cheriet, M. (1995). Financial document processing based on staff line and description language. IEEE transactions on systems, man, and cybernetics, 25(5), 738-754. DOI: https://doi.org/10.1109/21.376488

Tarawneh, A. S., Hassanat, A. B., Chetverikov, D., Lendak, I., & Verma, C. (2019). Invoice classification using deep features and machine learning techniques. Paper presented at the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). DOI: https://doi.org/10.1109/JEEIT.2019.8717504

Toğaçar, M., Cömert, Z., & Ergen, B. (2021). Intelligent skin cancer detection applying autoencoder, MobileNetV2 and spiking neural networks. Chaos, Solitons & Fractals, 144, 110714. DOI: https://doi.org/10.1016/j.chaos.2021.110714

Wang, G., & Gong, J. (2019). Facial expression recognition based on improved LeNet-5 CNN. Paper presented at the 2019 Chinese Control And Decision Conference (CCDC). DOI: https://doi.org/10.1109/CCDC.2019.8832535

Xia, Y., Cai, M., Ni, C., Wang, C., Shiping, E., & Li, H. (2019). A Switch State Recognition Method based on Improved VGG19 network. Paper presented at the 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). DOI:https://doi.org/10.1109/IAEAC47372.2019.8998029

Zou, Y., Zhao, L., Qin, S., Pan, M., & Li, Z. (2020). Ship target detection and identification based on SSD_MobilenetV2. Paper presented at the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC). DOI: https://doi.org/10.1109/ITOEC49072.2020.9141734