Enhanced Vehicle Classification Using Transfer Learning and a Novel Duplication based Data Augmentation Technique

A Thesis submitted to Tokushima University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

March 2022

Abdelrahman Maher Abdelkader Harras

Tokushima University
Graduate School of Advanced Technology and Science
Information Science and Intelligent Systems

Table of Contents

TABLE OF CONTENT

List of Tables

List of Figures

Acknowledgment

In the name of ALLAH, Most Gracious, Most Merciful. I thank ALLAH who gave me the strength and wisdom to do this work. I have many thanks to those who have offered me support, encouragement, and motivation throughout my thesis journey at Tokushima University.

First of all, my thanks go to my Research Director, **Professor Kenji Terada**, Professor in the Faculty of Science and Technology, Graduate School of Advanced Technology and Science, Tokushima University. I am grateful to him for his invaluable advice and his commitment throughout this work. His encouragement allowed me to overcome difficult times. I express my gratitude to him for his trust, his availability, and his responses to my many requests. It is a privilege that he granted me by being the director of this research.

Secondly, **Dr. Akinori Tsuji**, in the Faculty of Science and Technology, Graduate School of Advanced Technology and Science, Tokushima University, Japan. I learned a lot from him and nothing would have been possible without his many human and scientific qualities. His daily dedication, his profession, and his passion are the driving force of my efforts to complete this research. His smart devices have guided my research from the very beginning until its final results.

Third, **Dr. Stephen Karungaru,** in the Faculty of Science and Technology, Graduate School of Advanced Technology and Science, Tokushima University, Japan. Many thanks to him, not only for his extreme patience but also for his intellectual contributions to developing my thesis. Thanks to him for his kindness, his support, and his responsiveness. for his unwavering support throughout my work to finish the thesis.

Fourth, **My wife Pasant**, who has always believed in me and who has supported me in this thesis project throughout the many years of study. She has been always my major pillar of strength and my life backbone. I will never be enough grateful to her.

At last but not least, **MY parents**, the ones who can never be thanked enough, for the overwhelming love and care they gave me.

Abstract

Due to the traffic crisis seen in most of the large cities, intelligent transportation system (ITS) applications are widely installed to provide traffic management services. Vehicles' classification, which is the final step in the vehicle detection process, has shown to be significant in many of these applications, especially those whose main interest is to classify vehicles in the context of monitoring roads and maintaining their safety. For instance, in sensitive areas like airports, it is only allowed for specific vehicle types to park or move while other types are not allowed. That is why in such areas there is a need, not only to detect vehicles but also to categorize their types.

In this Ph.D. dissertation, we proposed a robust Deep Neural Network (DNN) based computer vision model to classify vehicles. In the vehicle classification process, objects are initially classified into two classes; vehicle class or non-vehicle class, and then classified according to their types i.e. Crossover, Sedan, Hatchback, pickup, Van, and Minivan, etc. In the proposed DNN based model, we used Transfer Learning and data duplication- based- augmentation to reach the optimum classification performance. Based on the Transfer Learning approach, a pre-trained ResNet-50 network was used to obtain a highly effective learning model. we removed the final layers of the network, forwarded the produced feature maps, which are the relevant parameters representing the characteristics of the vehicles, and allowed their identifications, to the classification layer that is preceded by connected layers. Afterward, we re-trained the model on a small dataset that includes only vehicle' categories of interest. Therefore, it learns only those categories' features during this re-training process without being confused by the features of other classes learned earlier. The problem encountered in this context is that an effective Deep Neural Network usually requires a large amount of data to train on but actually, there isn't as much data as needed. Hence, data augmentation techniques are implemented to increase, artificially, the amount of training data by using the existing data and, accordingly, improve the generalizability of the model. In the proposed method, a new approach for enhancing training data augmentation was proposed. In this approach, data augmentation by duplication was implemented through which the training dataset instance of each vehicle type was used side by side with their duplicates in every epoch. i.e. In the training process, each instance of each type of vehicle has duplicated multiple times in successive training sessions until the optimal learning results, that make the training process to be at its maximum performance, are reached.

ABSTRACT

We proved, empirically, that this technique of augmenting training data by duplication, enhances the classification performance by a considerable value. In the experiments, a testing dataset, including 640 images of 6 vehicle types was used to evaluate the proposed method. The model training was implemented in a staged manner. In the first stage, the training dataset has only the baseline instances of the vehicle dataset without any duplication. In every subsequent stage, the number of duplicates per image was increased by one after which the training was performed and the model was re-evaluated. The overall accuracy of the model was recorded in each stage. This is continued until the optimum classification accuracy has been reached.

To ensure the method generalization, we re-tested this optimum model using the Stanford-based custom dataset of 8,000 images and indicated that the model not only behaves properly with the testing dataset but also behaves similarly with a real-world dataset. We compared the proposed method with the existing ones and it was shown that it outperforms all of them.

From the experimental results, we proved that the overall classification accuracy has improved from 92.68 % without any duplications in the training dataset, to 99.70 % with 4 duplicates for every instance in the training dataset. In future work, it is intended to use the proposed method to enhance the fine-grained classification of vehicles.

Keywords: Vehicle classification, Data augmentation, Transfer learning, Deep Neural Networks

CHAPTER 1

Introduction

Vehicle classification is one of the solutions for the problem of traffic overcrowding through which large amounts of data from roads are collected and analyzed for proper decision-making. In general, these tasks are previously performed by a human, now they are changed to be fully automated in Artificial Intelligence (AI) applications. Deep Neural Networks (DNN), which is a form of AI application, are widely used in vehicle classification solutions. Vehicle type recognition is required as part of the automatic toll, where the system should be able to characterize which type of vehicles (car, van, truck, .etc.) go to the tollbooth to assess the price the driver should pay. It is also required in video surveillance applications used to monitor pollution peaks. In these applications, recognizing the vehicle's type allows for better regulation by setting up different routes according to their size, and accordingly, to their related pollution. Moreover, it is important to categorize vehicles according to their types in applications concerned with driving assistance, automatic parking, and autonomous driving. In this context, the estimation of the orientation and 3D localization of vehicles around the smart vehicle is required to predict its appropriate path and speed. Additionally, in security implementations, it is important to monitor and classify vehicles around the clock in sensitive areas, like airports, cantonment areas, and secure locations. In such areas, it is only allowed for specific vehicles to park or move while other types are not. In these implementations, a pre-trained DNN is used to automatically capture classes of interest in the upcoming stream of vehicles and make decisions accordingly. However, the generalization of DNN based models depends on the amount of data they have trained on. More training data is necessary to make a more robust learning model. It is a real challenge to get enough labeled data to train DNN models because it is of high cost and time-consuming. Increasing data artificially is one of the most suitable solutions for this problem as, instead of trying to find and label more data, we build new training data based on what we have. In our research, we investigate how to effectively increase the training dataset artificially by creating new data from the existing dataset. The most well-known

technique is data augmentation in which modified images are created from the available data or images. It is used in most DNN based solutions and with the most complex and powerful algorithms regardless of the data kind. In [1] for instance, plant leaf recognition is among those diversified DNN based implementations that use data augmentation techniques. The size of the plant leaf training dataset was increased by 25 times by using simple techniques like rotation, blur, contrast, scaling, illumination, and transformation.

The objective of our research is to develop a high-performance method to classify vehicles in an easy and fast way by training a pre-trained DNN on an effective dataset that is artificially enlarged by data augmentation. The proposed data augmentation technique is a new approach. The training dataset is augmented by duplicating its instances for getting highly effective training. In this work. Transfer Learning is used for changing the final layers of the pre-trained network. The final layers are removed and the networks are re-trained on a small amount of dataset including only vehicle classes of interest. Therefore, it learns only these classes' features without the features of other classes learned earlier. In addition, during the re-training process, the data augmentation technique is used to create image variations that improve the training performance and the ability of the trained model to be generalized. This technique is enhanced by duplicating vehicle image instances multiple times in the training dataset. Accordingly, the duplication assists the simple random augmentation such as cropping, flipping, and rotating in each epoch during the training. In the proposed method, the optimum duplication number, which generates the highest classification performance, was empirically identified.

This work contributes to enhancing vehicle classification in DNNs by implementing a novel technique for augmenting the training dataset by implementing multiple duplications to its instances. Thus, it transforms the data duplication phenomenon from being very negative to being a good one. This technique increases the training data efficiency and thus reduces significantly the performance gap between lab test results and real-world ones. Besides, in this work, a good example of using a limited number of objects to train a DNN model effectively, through using Transfer Learning, was demonstrated.

This research was inspired by the previous work in my Ph.D. program [2] through which how to detect various vehicles using an object detector was explored. Object detection is one of the most challenging areas of computer vision affecting people's life widely and it deals with detecting instances of visual objects of a certain class, such as vehicles, in digital images. i.e. image might contain multiple objects of different classes and the goal is to localize those objects, and for the object of maximum accuracy, we define the most likely class. That work addressed several challenges. The first one is the computational time which is particularly important in the design of

multi-class detectors due to a large number of classes of objects to be detected: a good detector is a detector that maximizes performance while being the fastest possible model. The second concern is the varying appearance of objects which can vary according to several factors: the occultation, the point of view from which the objects are observed, and the size of objects in the image. The third challenge was the variation in the appearance of regions that do not correspond to objects (background). A detector must be able to decide whether a region is a background or an object part, even in complex environments. That deep learning-based work involves learning the appearance of the object using the learning algorithms and then searching for the object directly in the image. To do this, there is an algorithm that does not use the notion of movement or video and, generally, does not memorize previous detections. This approach effectively allows detecting specific classes of objects. Indeed, when objects of interest are known in advance, it is possible to use a trained detector specifically for this task. Accordingly, we used the YOLO object detector, which is the first one-stage detector in the deep learning era, that was proposed in [3] in 2015 because it is an extremely fast real-time multi-object detection model. From its name YOLO "You Only Look Once", it can be seen that it applies a single neural network to the entire image through which the image is divided into regions, and, for each region, a bounding box and class probability are predicted simultaneously. To compensate for the longer computational times due to the increase in data size, ResNet-50 [4], which is a CNN model that was pre-trained on the ImageNet dataset, is used through the context of transfer learning for the feature extraction in the proposed model. The learning capability of this convolutional neural network-based model allowed a significant improvement in the performance of object detectors because it can learn robust and high-level feature representations of those objects. However, the power of generalization of CNN drops dramatically when it is re-trained on a small dataset. To face this challenge, data augmentation techniques are used to enhance the performance of detection and classification systems, where the dataset is augmented by geometric transformations that take into account representative changes in perspective, including scale change and rotation. The current research evolves from this work as it is concerned with image classification which implies assigning a specific class label, among a set of possible class labels to a vehicle.

The rest of the document is structured as follows:

- Chapter 2 titled “Background” where we investigate and discuss several topics and technologies which are, one way or another, relevant to our work. These are computer vision, machine learning, Deep Learning, data augmentation techniques, object detection, and object classification.
- Chapter 3 titled “Related work”, where we present the works related to our work. Namely

vehicle type classification and fine-grained vehicle classification.

- Chapter 4 titled “proposed method”, where we mainly focus on how the proposed method is going to solve the research problem.
- Chapter 5 titled “Experiments” where we describe the testing environment, the model training, and the staged training. We also present the training parameters and indicate every type of training.
- Chapter 6 titled “Conclusion and future work”, and highlighted the important findings in our work and shows how they have emerged from the whole work. In the end, we highlight the future work of this thesis. In particular, the use of augmentation by duplication in the fine-grained classification of vehicles.

CHAPTER 2

Background

In this chapter, we investigated and discussed several topics and technologies which are, one way or another, relevant to our work. It shows first how computer vision has become more and more dependent on interdisciplinary collaboration between different fields of Artificial Intelligence. It highlights the relationship between traditional computer vision techniques and AI-based ones and how scholars have different perspectives in this regard. Afterward, it investigates the field of Deep Learning which has got a special interest in recent years, due to the great improvement of Neural Network-based algorithms. It highlights the perspective which considers computer vision as a subfield of Machine Learning, which is considered, by its turn, an application of Artificial Intelligence (AI) through which systems are provided with the ability to learn and improve from experience automatically, without being explicitly programmed. Then, we described the concept of Transfer learning, which is used heavily in our work. We show how it is used in machine learning methods where a model developed for a task is reused as the starting point for a model on another similar task. Data augmentation, which includes techniques used to increase the amount of data artificially, is addressed thereafter. It is accomplished by adding slightly modified copies of already existing data or newly created synthetic data from existing data. The different methods of data augmentation have been presented and their specific usage has been highlighted. In the last section, we detailed the process of image classification and presented the main approaches for object classification, namely unsupervised classification and supervised classification. Some details about these two main approaches are presented. Measures to assess the performance of a classification system are presented at the end of that section.

2.1 Computer vision

From the early days of computing, approaching human intelligence with the help of a digital system appeared as an objective to be reached [5]. This concerns many areas among which is the ability of humans to perceive their environment based on complex data, such as sensory information, they can get. Sensory perception consists of capturing this information and processing them to provide a coherent representation of the environment in which we live. It is therefore not surprising that what is called “computer vision” becomes the predominant technology to handle such sensory information.

Computer vision is an interdisciplinary field. It deals with how we could use computers to acquire an understanding of digital images. It automates the tasks that could be fulfilled through human vision [6]. Fig. 2.1.1. shows a simple comparison between computer vision and human vision. It is well known that human vision is difficult to imitate as it relies a lot on the intuition of human visual perception and thus computer vision would not be as comprehensive as human vision. However, scholars are always seeking to have machines designed to see the world differently and better than the human eye. Despite human perception allowing us to perform tasks like recognizing objects in an almost insignificant way, but getting a machine to achieve the same level of recognition has proven to be much more difficult. For example, humans can correctly identify the color of an object under different lighting conditions. The exterior can be bright, or it can be dark. There may also be fog. But we can always tell precisely what are colors the objects around us. This means that our visual system automatically compensates for a large part of these environmental changes while on the other hand, it is extremely difficult to deal with this variability in computers. However, we can say that, in computer vision, computers learn what should be the most important things in scenes they are watching. Besides, while visual functions such as motion analysis or object recognition are effortlessly performed by humans, these functions are extremely complex to implement using a digital system. However, In recent years, although human performance has not been achieved, significant progress has been made in computer vision whose technology has diversified implementations in many modern and solutions. They vary depending on what the computer is trying to identify and it is currently expanded to identify text, pictures, or faces. These advances are on the one hand due to the emergence of new vision algorithms, always more efficient, on the other hand, due to the development of digital components such than the GPUs (Graphics Processing Units) which allow these algorithms to be executed. Those components indeed offer significant computing power, necessary for most of these algorithms. However, they are not suitable for mobile applications which have limited resources: memory, calculation, surface,

and energy. The design of computer vision systems, capable of interpreting the content of images captured and able to be contained in our pocket, is, therefore, a major challenge for researchers in computer vision. There are currently smartphone apps that allow people to identify and find information on different species of plants or help to identify animals. Additionally, in facial recognition systems, parts of the face are measured by facial biometric software measures. This includes things like the color and shape of the eyes, nose, mouth, and chin. These features are called nodal points. A geometric pattern of a person's face requires about 80 nodal points. There are also computer vision-based onboard systems that can be used in a wide range of applications such as autonomous navigation in self-driving cars. In such systems, there is a need to determine what the objects are. there is a need also, not only to detect objects but also to decide what action to be taken based on objects types and their situations. For instance, the car should stop if it recognized a stop sign while it should analyze where the objects are, what their dimensions are, and what they are doing in case that the object is a car or a person [7].

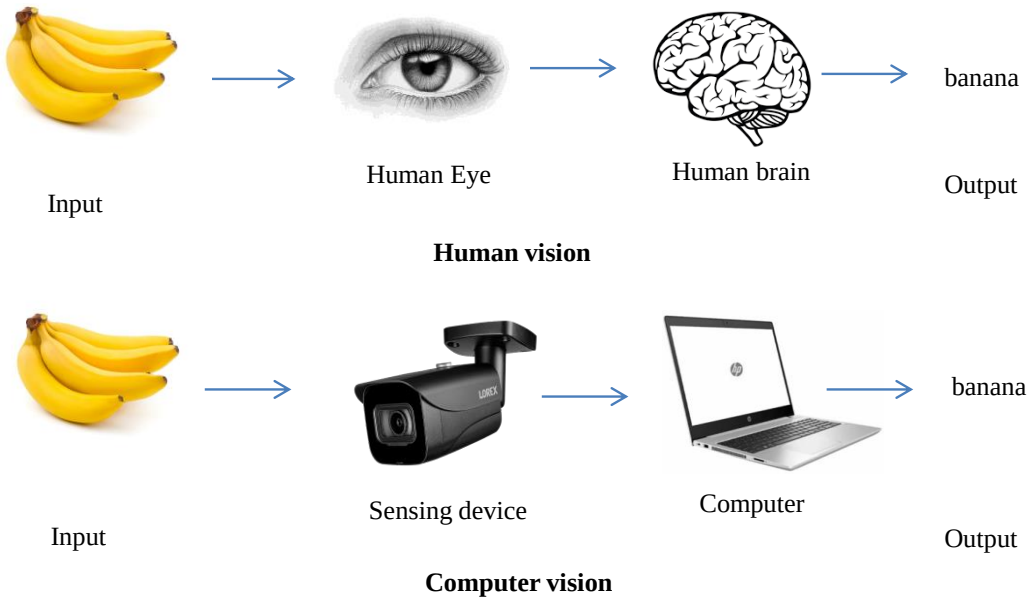

Fig. 2.1.1. Human vision vs. Computer vision.

Computer vision, which has always been collaborative, has become more and more dependent on interdisciplinary collaboration between different fields of Artificial Intelligence. We have seen significant changes in computer vision over the past few years. It can make a decisive contribution to complex robotics applications such as disaster response, including reconnaissance and mapping, survivor search, logistics, first aid, victim evacuation, and awareness support cooperative. However, vision systems of the unmanned ground vehicle (UGV) or unmanned air

vehicle (UAV) can produce gigabytes of images [8]. Added to this are the limitations on the quality of the images received, the diversity of objects of interest, and the unstructured surroundings [9]. Hence, new Artificial Intelligence tools are needed to identify useful information. In this sense, Convolutional Neural Networks (CNN) and Deep Learning can deliver effective results both for the visible spectrum and for the thermal infrared (TIR) [5]. This field of Deep Learning has got a special interest in recent years, due to the great improvement of Neural Network-based algorithms. It is a subfield of machine learning that involves processing multiple levels of signals and through which the world is hierarchically represented as nested concepts [10]. Machine learning, by its turn, is an application of Artificial Intelligence (AI) that provides systems the ability to automatically learn and improve from experience rather than programming algorithms. In machine learning, the focus is on developing computer programs capable of accessing data and using it in self-learning [11]. Fig. 2.1.2. illustrates that there are different perspectives regarding the relationship between the traditional computer vision techniques and the different AI-based ones and how far the computer vision is interrelated with them.

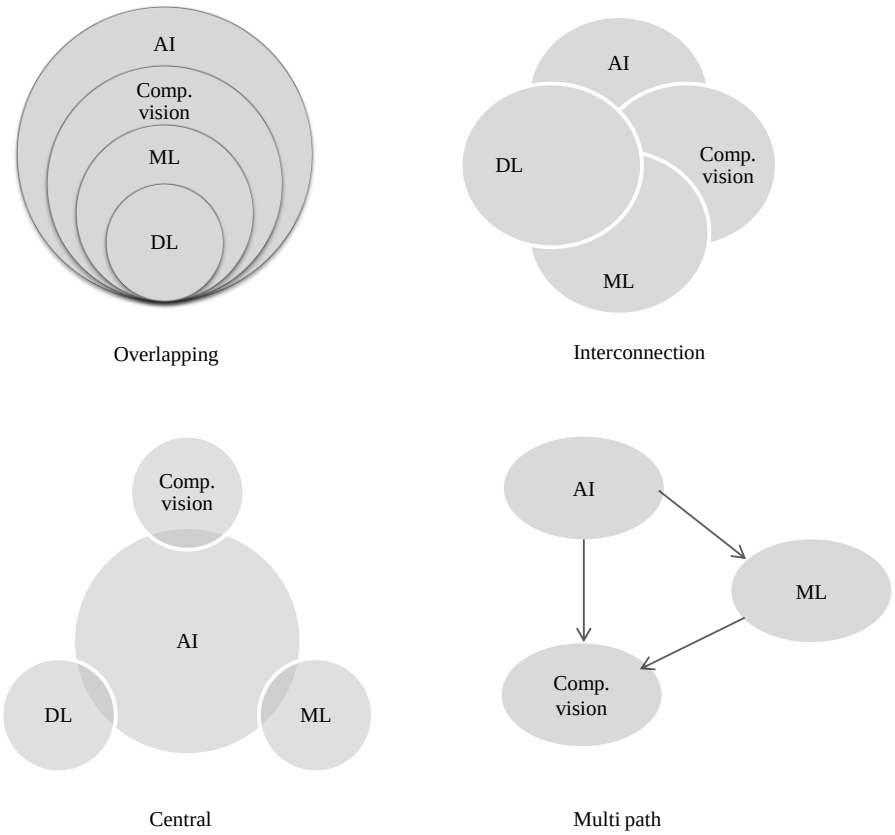

Fig. 2.1.2. different perspectives regarding the relationship between computer vision and different AI techniques.

Currently, we could achieve state-of-the-art results on challenging computer vision problems, like object detection and recognition, image classification, and face recognition, by using Deep Learning. It has become an obvious choice for many scholars who see that learning what constitutes the human visual system gradually became a huge source of inspiration for them. For instance, our eyes are constantly exploring our field of vision and so are computer vision systems. However, a machine that has to follow an individual through an outdoor video will probably be able to safely ignore the sky and focus on the lower part of the image. In Deep Learning techniques, what are scholars doing is replicating the different types of attention patterns needed for this type of scenario. They use a collection of manual tracking data that we gathered while different people were looking at images. These data are added further to a Deep Learning algorithm through which a Convolutional Neural Network is trained to replicate the attention patterns. They studied how this work can be applied to systems that help authorities control toll and carpooling mechanisms. The developed systems help determine how many passengers are traveling in a car by eliminating irrelevant background information to improve systems accuracy. They also developed augmented reality apps that would be particularly useful for people who drive vehicles they are not familiar with. The mobile application allows users to scan the interior of a car and it pops up on the screen the function of each element of the dashboard and the attention model can be trained to locate those areas and enable the app to quickly focus on the elements of interest. However, a problem confronting the adoption of CNNs in new applications is the scarcity of conveniently labeled images needed for application cases or specific modalities. In this sense, knowledge transfer techniques are effective in different applications [12] and can be combined with operations to increase data to get good results from small data sets.

2.2 Deep learning

Deep Learning, as mentioned earlier, is a branch of machine learning, which is by its turn a branch of AI. It dates back to 1943 when a computational model for Neural Networks based on mathematics and algorithms, called threshold logic – was created to mimic the process of thought of the human brain. Since then, Deep Learning has evolved steadily [13]. In 1979 a multi-layered Artificial Neural Network, called Neocognitron [14], was developed. It was capable of learning to recognize visual patterns. The second break in the development of Artificial Intelligence took place in (1985-1990), which also made it possible to research Neural Networks and Deep Learning. In 1995, the support vector machine (SVM) was developed [15]. It is a supervised machine learning algorithm capable of performing classification and recognition tasks. It can be used for text

categorization, handwriting recognition of characters, and image classification for machine learning and Deep Learning.

The next big step in the evolution of Deep Learning took place in 1999 when Graphics processing units (GPUs) have been developed and thus computers began to speed up data processing. This made the computational speeds 1000 times faster over 10 years. Meanwhile, Neural Networks began to compete with support vector machines and started to offer better results using the same data. Neural networks also are capable of continuing to improve as new learning data are added. Around the year 2000, the problem of the evanescent gradient appeared. It had been discovered that the characteristics formed in the lower layers were not learned by the upper layers, because no learning signal reached them. This was not a fundamental problem for all Neural Networks, it was limited to learning methods based on gradients. Two solutions were used to solve this problem: layer-by-layer pre-learning and the development of short-term memory.

In 2009, the NIPS workshop working on Deep Learning for voice recognition discovers that with a sufficiently large data set, Neural Networks do not need prior learning, and error rates drop considerably. Hence the emergence of the ImageNet database, which is a free database of over 14 million labeled images, has been proposed [16]. Since the significant increase of GPUs by 2011, it becomes possible for convolutional to work without a layer-by-layer algorithm. With the increase in computing speed, it became evident that Deep Learning had considerable advantages in terms of efficiency and speed. AlexNet [17] that has won several international competitions in 2011 and 2012, is an example. It initially contained only eight layers; five convolutional layers followed by three fully connected layers, and have a boosted speed and a rectified linear unit (ReLU). Its success launched CNNs on a path to success in the Deep Learning community.

2.2.1 Artificial Neural Networks

An Artificial Neural Network is the basis of Deep Learning. The basic unit of computation in an artificial Neural Network is the neuron which was defined in [18]. An artificial neuron receives inputs from some other neurons or an external source having numeric values x_1, x_2, ..x_n to which it is connected by synapses and calculates an output. Each entry x_i has an associated weight w_i, which is assigned based on its relative importance compared to the other entries. The input value x of the neuron corresponds to the weighted sum of its inputs by adding another entry with a weight b called bias. Then, the neuron applies a function f to this sum, as shown in Fig. 2.2.1. The output of the neuron y, called activation output, is calculated according to formula (2.2.1):

$$y = f\left(\sum_{i=1}^{n} wi\, xi + b\right) \tag{2.2.1}$$

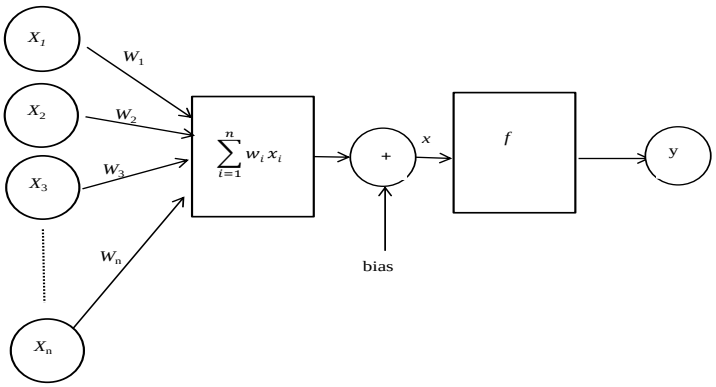

Fig. 2.2.1. Modeling of an Artificial Neural Network.

where *f* is the activation function.

The forward propagating Neural Network was the first and simplest Artificial Neural Network to be developed. It contains several neurons arranged in layers. Neurons in adjacent layers have connections between them. All of these connections have weights associated with them. An example of a forward-propagating Neural Network is shown in Fig. 2.2.2.

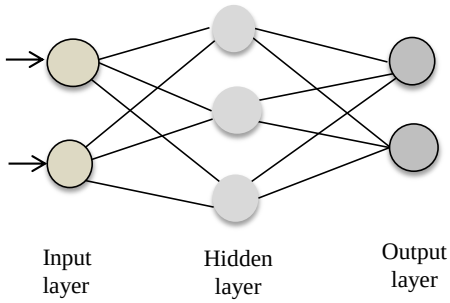

Fig. 2.2.2. Example of an Artificial Neural Network.

As shown in Fig. 2.2.3**.**, a forward propagating Neural Network can be made up of three types of neurons:

- Input neurons provide information from the world outside the network and together build an "input layer". No calculation is performed in the input nodes, the information is only passed to hidden nodes.
- Hidden neurons: hidden neurons have no direct connection to the exterior world (hence the name "hidden"). They perform calculations and transfer information from input neurons to output neurons. A hidden layer is formed from a set of hidden neurons. Although a forward-propagating network had historically only one input layer and one output layer, it can have no or more hidden layers, according to the principle of generalization.

- Output neurons: the output neurons together build an “output layer “.They are responsible for the calculations and the transfer of network information to the outside world.

In a forward-propagating network, information moves from the input neurons, passing through the hidden neurons (if they exist), and arriving at the output neurons. A Neural Network is organized as an acyclic-oriented graph. Two examples of “forward propagation” networks are presented in Fig. 2.2.3.

- Single-layer perceptron SLP- Which is a Neural Network consisting of a single artificial neuron having a binary output value which is determined by applying a monotonic function to a linear combination of the input values, with error correction learning. It is a linear classifier, and in the case of nonlinearity, the learning process will never reach a point where all cases are classified correctly.
- Multi-layered perceptron – It has one or more hidden layers where each layer consists of several varying neurons. It is a type of formal Neural Network that is organized into several layers. As information flows from the input layer to the output layer, it is thus a feed-forward network. It makes it possible to classify nonlinear groups. It is therefore solves problems that single-layer perceptron cannot solve.
- The function *f*, called the activation function, is non-linear. Its purpose is to introduce non-linearity in the output of a neuron. This is important because most real-world data is nonlinear and therefore neurons will learn these non-linear representations. It should also be differentiable to be able to apply the gradient back propagation algorithm. Each activation function takes a unique number and performs some mathematical operation on it. There are many activation functions used in practice

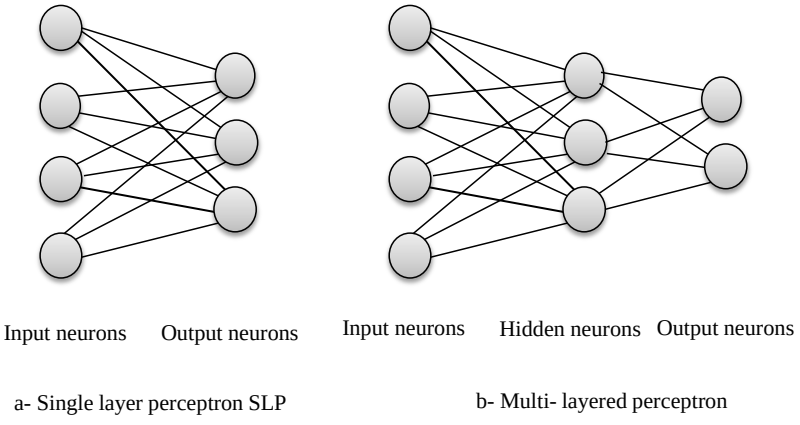

Fig. 2.2.3. Two types of perceptrons.

- The Sigmoid function: takes a real input and reduces it between 0 and 1

$$f(x) = \sigma(x) = \frac{1}{1+e^{x}} \quad (2.2.2)$$

- The hyperbolic tangent function: takes an input of real value and reduces it to a value in [-1, 1]

$$f(x) = \tanh(x) = 2\sigma(2x) - 1 \quad (2.2.3)$$

Fig. 2.2.4. illustrates both functions.

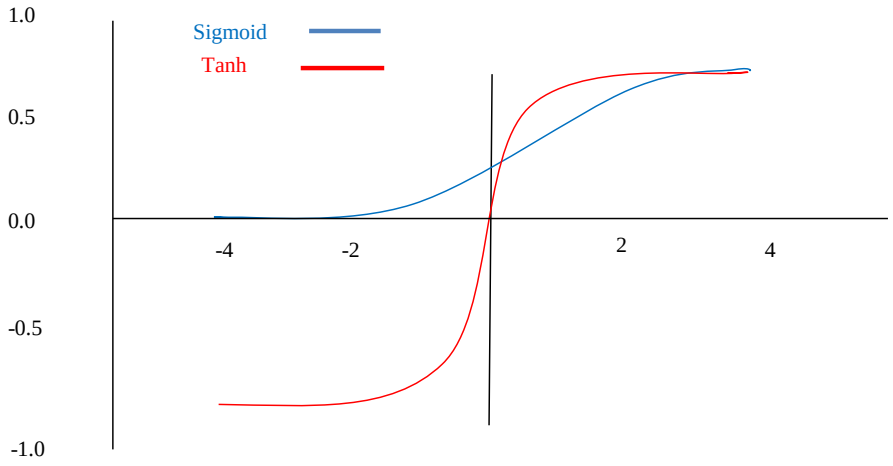

Fig. 2.2.4. Sigmoid and hyperbolic functions.

During the learning phase, the calculation of the gradient at each iteration allows having an exponential decrease in the gradient. In the opposite direction, there may be explosive gradients where the value exhibits exponential growth. To avoid this problem of evanescent gradients, the use of nonlinear activation functions and non-saturating (functions that do not have dwellings) is recommended. For this, an additional operation called ReLU (Rectified Linear Unit) was used after each convolution operation thus showing optimization of the Artificial Neural Networks. Note that this function is non-differentiable but the procedures could be adapted to be able to use it. The ReLU function is an elementary operation (applied by element) and replaces all negative values by zero. It is defined as in formula 2.2.4

$$f(x) = \max(0,x) \quad (2.2.4)$$

Compared with sigmoid and tanh, the ReLU function speeds up the learning time which results in better computational efficiency [19]. There are studies in applied mathematics that are interested in the expressiveness of Neural Networks. Indeed, it is a question of studying the capacity of these networks to represent certain classes of functions. Research has shown that the capacity is

proportional to the depth of the network as well as to its width [20]. The depth of a Neural Network is defined by the number of layers it contains and the width by the number of neurons on the layers.

The learning phase consists of adjusting the weights of a Neural Network to obtain the best regression or classification results. The goal of this optimization is to minimize the cost function Ψ in (2.2.4) which computes the error between the data from the learning base *y* and the data predicted by the Neural Network ẏ.

$$\Psi(p) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \dot{y}_i)^2 \quad (2.2.5)$$

According to [21], one of the optimization algorithms proposed is the back propagation algorithm which is based on gradient descent, through which it is possible to calculate the gradient of a network, based on another neural network assisting the original one as it is known as the learning rate and it is considered one of the important hyper-parameters for the optimization of Deep Neural Networks by acting on its convergence. Indeed, too high a learning rate leads to important weight updates and convergence becomes unstable. On the other hand, for a low learning rate, convergence is slowed down with the possibility of falling into local minima. The popular approach used in Deep Learning to have the optimal learning rate is to start learning with a high value to accelerate the gradient descent and then reduce it to improve accuracy.

Among the problems that can be encountered during the learning process is over-fitting. This is due to the incorrect setting of the gradient descent hyper parameters or due to database issues. In this regard, the Neural Network captures information that is not useful to accomplish its task or becomes unable to generalize the f of the data, which limits its ability to recognize new data. To avoid this problem of over-fitting, regularization techniques have been proposed. The first technique is the Dropout [22]. This involves eliminating certain neurons at each iteration to decrease the number of network parameters and update a reduced number of weights. The second technique is data augmentation. The idea is to increase the number of training samples by adding some variety without affecting their semantics. For example, we can add images by introducing rotations, vertical symmetry, or even changing their dimensions.

2.2.2 Convolutional Neural Networks

Convolutional Neural Network (CNN) is a special type of Artificial Neural Network explained earlier. It was introduced in the early 1990s [23]. CNNs have their “neurons” arranged like the neurons of the front lobe, which is the area responsible for processing visual perception in humans and animals [24]. Their name reflects one of the most important operations in the network:

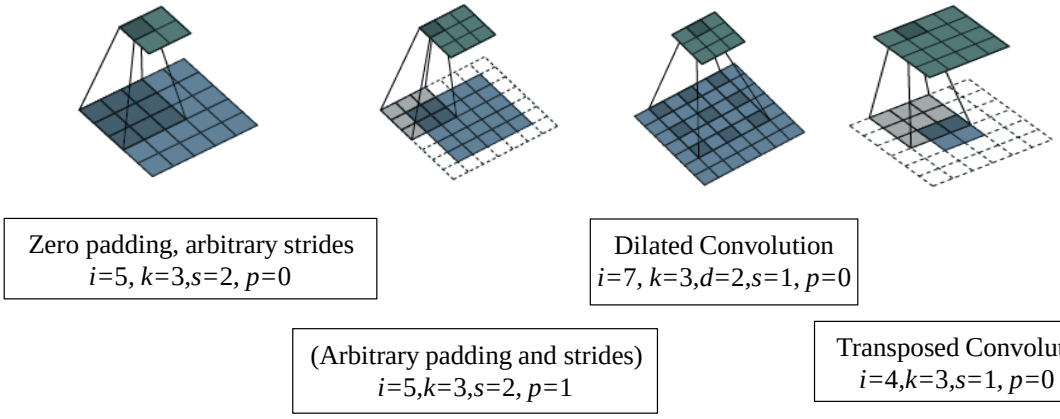

Fig. 2.2.5. Different types of convolution [25].

convolution [26]. There are different types of convolutions. Fig. 2.2.5. shows examples of these types. Convolution handles three important concepts that can improve a machine learning system:

1- Convolutional interactions: In a traditional Neural Network, the layers use matrix multiplication by a matrix parameter matrix with a separate parameter describing the interaction between each input unit and each output unit. However, convolutional networks generally have sparse interactions. This is accomplished by making the core smaller than the entrance. For example, the input image can be thousands or millions of pixels in size, but with convolutional layers, we can detect useful small features, such as outlines with cores occupying only tens or hundreds of pixels. This allows fewer parameters to be stored, which reduces the need for memory and improves the efficiency of the algorithm. It also means that calculating the output requires fewer operations.

2- sharing of parameters: It consists of using the same parameter for several functions in a model. In a traditional Neural Network, each element of the weighting matrix is used only once when calculating the output of a layer. It is only multiplied by one element of the entry. Sharing parameters allows to have related weights, that is to say, the value of the applied weight to an entry is linked to the value of another weight applied elsewhere. Sharing parameters used by the convolution operation means that instead of learning separate sets of parameters for each location, we learn them together. It does not affect the execution time of “forward propagation” but it further reduces model storage requirements.

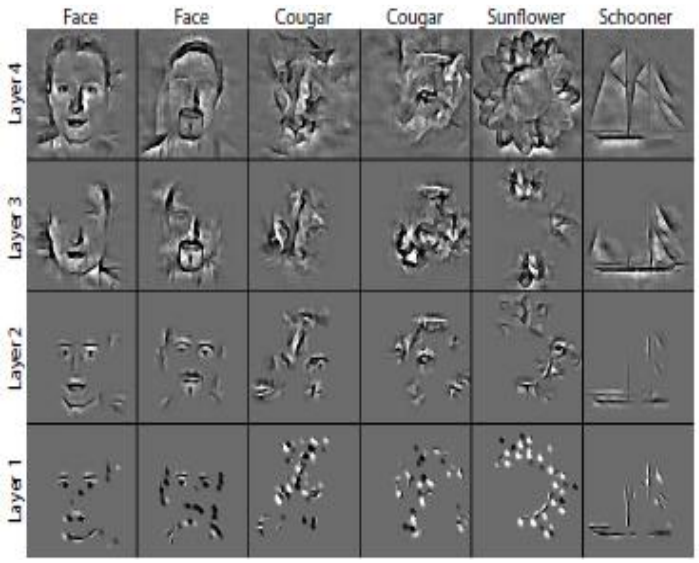

Fig. 2.2.6. The deeper the CNN, the more comprehensive features we get [27].

3- Equivariant representation: This means that if the input changes, the output changes in the same way. Otherwise, a function $f(x)$ is equivalent to a function $g(x)$ if $f(g(x)) = g(f(x))$.
In the case of a convolutional layer, a translation of the input image causes that activation maps to be translated. In [28], A convolution layer at the start of a neuron network was introduced to extract images' features in a relevant way through the convolutions' nucleus. Note that the relevance of features is proportional to the late intervention of convolutions. That is, the more the convolution occurs later, the more complex the nuclei become and the more its capability of detecting more details. i.e. The deeper the network, the more details we get as shown in Fig. 2.3.6. Indeed, at the first convolution layer, the values of a kernel are initialized randomly. Then these values will be updated with the network learning process to improve feature extraction results. Further details of the Convolutional Neural Network are presented in Chapter 4.

2.3 Transfer learning

Humans could recognize and apply relevant knowledge from previous learning experiences when we come across new tasks. The more related a new task is to our previous experience, the more easily we can make use of the previously acquired knowledge. In machine learning, the Transfer Learning technique allows the transfer of knowledge from one domain to another makes machine

learning as efficient as human learning. Thus, the Transfer Learning objective is to make use of the acquired knowledge learned in one or more previous tasks to enhance learning in a new task in the context of machine learning and Deep Learning. In other words, Transfer Learning is a method of machine learning through which we reuse a model, that developed for one task, as a starting point for a model in another task. It refers to all the methods through which the knowledge gained from solving problems is transferred to treat another problem. From another perspective, in the context of generalization, as proposed by [29], Transfer Learning is the situation in which what is learned in an environment is used to improve generalization in another environment.

2.3.1 Transfer learning mathematical framework

Authors in [30] use the domain, the task, and the marginal probabilities to present a framework for understanding Transfer Learning. The framework is defined as follows:

- A domain, D, is defined as a two-element tuple type consisting of the feature space X, and the marginal probability distribution P(X). Therefore, we can represent the domain mathematically as in (2.3.1)

$$D = \{ X, P(X)\} \tag{2.3.1}$$

  Where:

  - Feature space: χ
  - Marginal probability distribution: P (X)
  - $X = \{x_1,\ldots, x_n\}$, $x\ i \in \chi$

  Here x_i represents a specific vector.

- A task T is defined as a two-element tuple type of the label space Y, and the objective predictive function $f(x)$, which can be denoted as P (y | x) from a probabilistic point of view.

For a given domain D , a task is defined by two components as in (2.3.2)

$$T = \{Y, P(y \mid x)\} = \{Y, f(x), \} \tag{2.3.2}$$

- where $Y = \{y_1,\ldots, y_n\}$, $y\ i \in \gamma$
- A label space: Y
- A predictive function $f(x)$ learned from feature vectors or label pairs.
- For each feature vector in the domain, $f(x)$ predicts its corresponding label: (x_i, y_i)

Therefore, using these definitions and representations, we can define Transfer Learning as follows: When a source domain is DS, a corresponding source task is TS, and a target domain is DT and a target task is TT, the goal of Transfer Learning is to have the performance of the predictive

function $f_T(.)$ improved in the learning task TT by the information obtained from DS and TS where DS ≠ DT or TS ≠ TT. In most cases, The size of DS is assumed to be much larger than the size of DT, and that a limited number of labeled target samples NT are available, which is exponentially less than the number of labeled source samples NS (i.e. NS≫ NT)

2.3.2 Perspectives of Transfer Learning scenarios and configurations

Transfer learning is useful in image and video analysis tasks. In addition, this approach very often gives satisfactory results, in particular when the training data set is relatively small. It has exhibited great success with the rise of Deep Learning. Indeed, very often, the models used in this field require high computation time and significant resources. However, using pre-trained models as a starting point, Transfer Learning makes it possible to quickly develop high-performance models and effectively solve complex problems in many areas among which is computer vision. It is based on a simple idea, that of re-using the knowledge gained in other configurations (sources) for solving a particular problem (target). In this context, we can distinguish several approaches depending on what we want to transfer, when and how to carry out the transfer. Overall, according to [31], we can distinguish three types of Transfer Learning as demonstrated in Fig. 2.3.1.

1- **Inductive Transfer Learning.**

The goal of inductive Transfer Learning is to have better performance in the target predictive function. In this configuration, source and target domains similarity does not matter, while the source and target tasks are different but close. The idea is then to use the existing models to advantageously reduce the scope of the possible models (model bias). For example, it is possible to use a model trained for detecting animals to build a model capable of detecting dogs. In this configuration, the labeled data is available in DT and the target task TT is different from the source task TS whatever the type of source domain DS and the target domain DT is. The transfer in inductive learning is achieved by allowing knowledge of TS to improve the generalization of TT. If the source domain is similar to the target domain, then the learning speed of the model will be improved due to the freedom to choose and adjust the inductive bias TT according to the knowledge TS. Inductive Transfer Learning can be categorized further into two cases:

Case 1. No available labeled data in the DS. It is thus similar to the setting of the self-taught learning where there might be a difference between the label spaces of DS and DT which is equivalent to the unavailability of labeled data in the DS in inductive T.L.

Case 2. There is a lot of labeled data in the DS. It is thus similar to the setting of multi-task learning. However, inductive T.L focuses only on improving the performance of the TT through transferring knowledge from DS, while in multi-task learning, both source and target tasks are learned simultaneously.

2- Transductive transfer Learning.

In this configuration, the source and target tasks are similar, but the corresponding domains are different either in terms of data or marginal probability distributions. For example, NLP models, such as those used for morphosyntactic word tagging, Part-Of-Speech Tagger (POS Tagger), are typically trained and tested on topical data such as "new york times". They can be adapted to data from social networks whose content is different but similar to that of newspapers. In this configuration, the source task TS and the target task TT are identical, but the source domain Ds and the target domain DT are different. Here, most of the labeled data exist in the DS, and there is no labeled data present in the DT. The 2 scenarios that could be observed in transductive Transfer Learning are:

(1) The existence of different feature spaces in DS and DT (i.e. $XS \neq XT$);

(2) The existence of the same feature spaces in DS and DT (i.e. $XS \neq XT$), but with different marginal input probability distributions (i.e. $P(XS) \neq P(XT)$). This is related to what is called "domain adaptation" through which knowledge is transferred in the context of text classification.

3- Unsupervised Transfer Learning

As with inductive Transfer Learning, the source, and target domains, DS and DT are similar, but the tasks are different. The data in the two domains are not labeled. It is often easier to obtain large amounts of unlabeled data, from databases and web sources for example, than labeled data. This is why the idea of using unsupervised learning in combination with Transfer Learning has got a great interest. For example, self-taught clustering, which is a case of unsupervised learning, is an approach that enables the clustering of small collections of unlabeled target data, with the help of a large amount of unlabeled source data. This approach proves to be more efficient than the advanced approaches traditionally used when the target data is labeled in an irrelevant way[32].

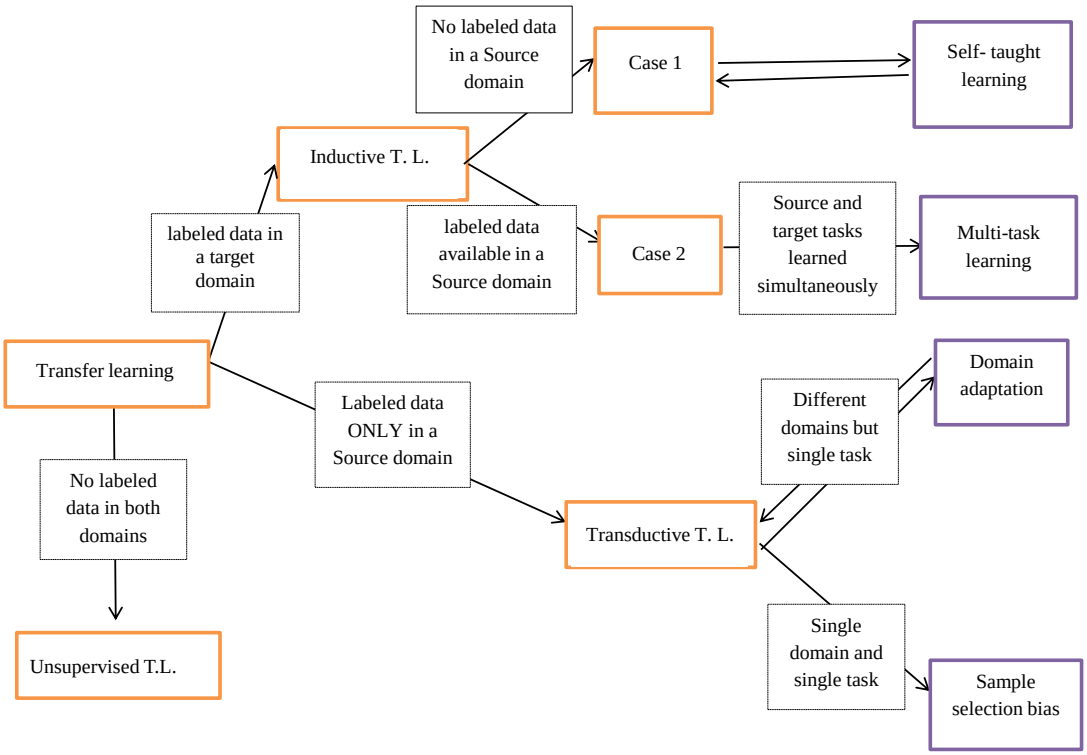

Fig. 2.3.1. An overview of Transfer Learning types [33].

2.3.3 Transferable components in Transfer Learning

Table 2.3.1. depicts transferable components in various Transfer Learning types. According to [34], the following describes what should be transferred through the previously presented categories of Transfer Learning:

- **Instance transfer**: the reuse of knowledge from the source domain to the target task is usually an ideal scenario. In most cases, data from the source domain are not used directly. Rather, there are certain instances of the source domain that can be reused in the target data to improve results after being re-weighted.
- **Feature-representation transfer**: Its goal is to minimize the divergence of domains and reduce feature error rates by identifying their good representations to be used from source to target domains. Depending on the availability of labeled data, supervised or unsupervised methods can be applied in the context of Transfer Learning based on feature representation.
- **Parameter transfer**: this approach works under the assumption that source and target tasks share some parameters or share prior hyper-parameters distributions. Thus, knowledge is transferred across tasks through those shared parameters.

- **Relational-knowledge transfer**: unlike the previous three approaches, the transfer of relational knowledge handles data that is not independently and identically distributed, but rather there are some relationships among data elements. Such relationships are similar, in both source and target domains, so that they could be transferred.

Tab. 2.3.1. Transferable components in various Transfer Learning types.

	Instance T.	F. representation T.	Parameter T.	R. knowledge T.
Inductive T.L.	√	√	√	√
Transductive	√	√		
Unsupervised		√		

2.4 Data Augmentation Techniques

To address the lack of learning data, several methods for artificially producing new examples of data were used. The effect of increasing data is simply because it gives redundancy that helps to learn. It is then possible to carry out training on sets containing real and synthetic data in controlled proportions. The objective is to extend the training databases without not only losing representativeness but also respecting this constraint. The notion of control (mastery and knowledge) of the transformations applied is also important for a better understanding of the effects produced by the final system after learning. For instance, in the context of vehicle classification, CNNs have shown remarkable precision and competitive reliability compared to traditional methods. However, according to [35], CNN-based approaches require that the network be driven over a large amount of data. One of the main problems is that this quantity is not always available (data missing, not accessible, or too expensive). The artificial increase in data was introduced to resolve this problem and has become one of the best practices for improving the performance of CNNs. The traditional increase in the case of image data is based on basic transformations which generate samples that are extremely close to the distribution of the initial data [36]. There are several methods of data augmentation shown in Fig. 2.4.1. among which is the application of geometric transformation-based ones such as cropping, scaling, rotating, mirroring, and others on the images in the learning dataset, and are always combined for the sake of realism. This approach allows the CNN model to learn more diverse image features and therefore be able to correctly predict the category of the captured image.

For instance, augmentation by rotation is done by rotating the image to the right or left around

an axis at an angle between 1 ° and 359 °. Scale changes are made by multiplying the dimensions of the image by a factor, which thus allows either to enlarge the scale of the image (factor greater than

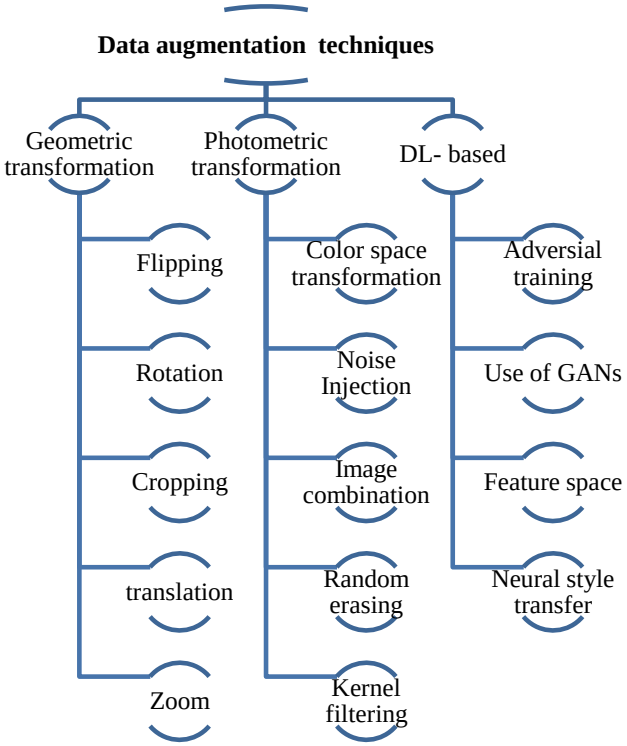

Fig. 2.4.1. Data augmentation techniques.

1) or to shrink it (factor less than 1). Note that CNN takes a fixed image size as well as a defined number of input channels and therefore all the images in the dataset are preprocessed then the change of scale is applied. Therefore, if the image is enlarged, parts of the edges will be eliminated and, if the image is narrowed, an outlined black will be automatically added. In the preprocessing operation for indoor scenes, for instance, dataset images are resized to 227 × 227 pixels and undergo color preprocessing if necessary (change from gray-level representation to color representation) to comply with the CNN image size and the number of channels which is equal to 3. It is worth mentioning that some geometric transformations may be suitable to some tasks while being not suitable to others. For instance, when objects are all of the same sizes and photographed at the same distance, a scale change is not applied to them. In addition, these techniques do not make it possible to produce a large number of items and result in databases whose representativeness is low compared to those techniques exclusively made up of acquisitions. However, they allow at the same time an excellent control of the synthetic data which are moreover almost indistinguishable from the acquisitions. More details on specific geometric-based data augmentation techniques follow.

1- Rotation.

Rotation-based augmentation is done by rotating the image to the right or left on an axis between 1st and 356o. The safety of rotation-based augmentation is strongly determined by the degree of rotation parameter. Light rotations, such as between 1^0 and 2^0, can be useful in digit recognition tasks like MNIST, but as the degree of rotation increases, the data label is no longer preserved after the transformation [37]. Fig. 2.4.2. demonstrates sample images rotated at right angles.

Fig. 2.4.2. Right angle rotation.

2- Cropping.

Cropping images can be used as a practical processing step for an image through which the central area of each image is clipped.

Cropped Original

Fig. 2.4.3 Cropping from right.

In addition, random clipping can also be used to provide an effect very similar to translations.

The difference between random cropping and translations is that in cropping, the size of the input is reduced (for instance from (256, 256) to (224, 224)), while in translation, the spatial dimensions of the image are preserved. Depending on the limit of reduction chosen for the cut, this may not lead to preserving labels. Fig. 2.4.3. shows examples of images cutting

3- Zoom.

A random zoom is obtained by the zoom range argument. A zoom smaller than 1.0 zooms in on the image, while a zoom greater than 1.0 zooms out the image, as we can see in Fig. 2.4.4. Additionally, flipping and translation are also among the geometric-based data augmentation techniques. See their details in section 4.

Fig. 2.4.4. Zoom example.

2.5 Image classification

Classification is one of the most common decision-making tasks in human business. A classification problem occurs when an object needs to be assigned to a group, or a predefined class, according to some observed attributes linked to this object [38]. Image classification involves assigning a label to an image from a predefined set of classes. Concretely, this consists in analyzing an input image and returning a label that classifies this image. The label always comes from a predefined set of possible classes. In machine learning, classification systems allow features to be extracted from images using specific algorithms such as SIFT. Thus, it is essential in machine learning to harness a set of extracted features for the system to learn. We can divide the process of image classification, using machine learning techniques, into three stages: data acquisition, data preprocessing, and classification decision. In the first step, data is acquired and preprocessed. Then, in a second step, the system extracts the features and also reduces their dimensions. In this regard, there are many well-known feature detectors like Local Binary Pattern, Scale Invariant Feature Transform, Accelerate-KAZE, Speeded Up Robust Features, etc. The last step is the classification where the trained classifier assigns the input image to one of the classes of images according to the extracted features. After converting an image to a feature vector, a classification algorithm will take this vector as input and display a class label. There are two main approaches for object classification: unsupervised classification and supervised classification. Some details of these two approaches are discussed in the following sections.

2.5.1 Unsupervised classification

Unsupervised learning also called learning from observations or discovery, constitutes determining a "sensible" classification from a set of given objects or situations (unlabeled examples). Unsupervised learning algorithms use a set of data containing many characteristics, then learn useful properties of the structure of this dataset. The majority of supervised non-learning algorithms are based on the clustering method which consists of dividing the dataset into groups of similar examples. We have a mass of undifferentiated data, and we want to know if they have any grouping structure, i.e. identifying a possible tendency for the data to be grouped into classes. From another perspective, in the unsupervised approach: no a priori knowledge; the classes are created automatically by the software. The classes are then named or labeled posteriori. This form of classification has existed since time immemorial. It concerns, in particular, the natural sciences, classification of documents, and also classification of sciences developed over centuries by philosophers. The automation of classification is a real field of research today. The key notion used to create object classes is a measure of the similarity between objects. Classes or concepts are constructed in such a way as to maximize intra-class similarity and minimize inter-class similarity. Unsupervised learning also corresponds to conceptual classification, where a collection of objects forms a class if this class can be described by a concept, given a set of predefined concepts [39]. Two unsupervised classification algorithms are commonly used: K-means and Fuzzy c-means.

2.5.2 Supervised classification

This approach requires a learning phase on the representative sample to learn the characteristics of each class and another phase to decide whether an individual belongs to this or that class. Accordingly, in this human-guided classification, there is a set of labeled data, or examples, that have been associated with a class by an expert and thus considered as a representative of this specific class. This set of examples constitutes the learning base that directs the image classification software to classify similar unknown samples. i.e. extrapolating these samples to other unknown samples. In supervised classification, a priori knowledge is used for the creation of classes. Thus, it is an inductive inference method that starts from a specific knowledge observed on certain objects and an initial inductive hypothesis and obtains an inductive assertion of observations to be related to specific classes and, accordingly, assigns a class to each object. The strong implication is satisfied if the algorithm correctly classifies all known objects. Some of the major supervised–based classification techniques are Bayes, the K nearest neighbors, and support vector machine. Supervised learning algorithms, like those mentioned earlier, use a dataset that contains

characteristics, but each example is also associated with a tag or tags.

In general, unsupervised learning involves observing several examples of a random vector x and implicitly or explicitly attempts to learn the probability distribution p (x), or some interesting properties of this distribution. While that supervised learning consists of observing several examples of a random vector and the associated value y and learning to predict y from x, generally by estimating p (y|x). The term supervised learning derives from the sight of the target provided thereby an instructor or supervisor who guides the learning process (hence the name supervised learning). Among the popular supervised classification methods: decision, logistic regression, support vector machine, K-nearest neighbors, and Deep Learning-based ones. Some details regarding the Deep Learning based methods follow in the next sub-section. Further details are shown in chapter 4.

2.5.3 Image classification using Deep Learning

It is one of the supervised classification methods. However, It has been entitled as a separate section because it has acquired a significant interest in classification problems due to its robust capabilities. In the past few years, Deep Learning-based solutions offer surprisingly better classification accuracy. In these solutions, Convolutional Neural Networks are used to automatically extract features and classify objects. The first layer of a CNN will learn small details of the image; subsequent layers will combine the previous features to create more complex information. In CNN, the feature extraction is performed by using filters. The network applies filters to the image to see if there is a match i.e. the item's shape is the same as part of the image. The large size and quality of the samples used in training allow the generation of robust Deep Learning models. These models require millions of parameters to train on, which limits the implementation of these approaches in many practical cases. Besides, this type of learning requires the availability of efficient hardware with processors with better and faster processing capacity and computing power. In addition to this, GPUs have also been very useful in computing millions of matrix operations at scale, which is the most common operation in any Deep Learning model.

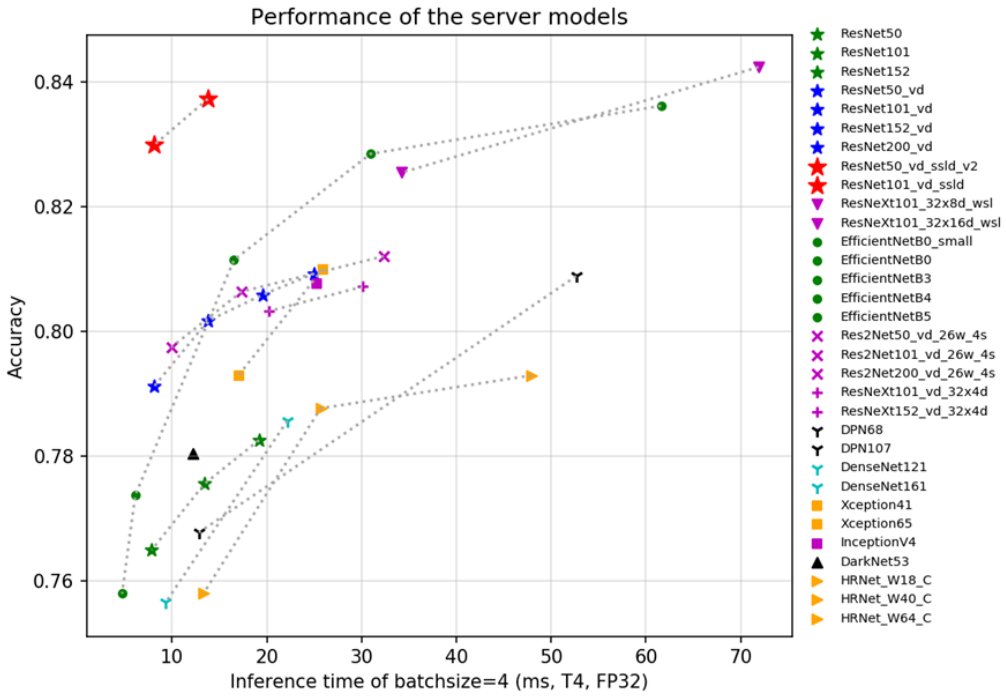

Fig. 2.5.1. Performance comparison chart between the known classification networks that have been trained on ImagNet classification dataset [40].

In 2009, the ImageNet database, which contains 14 million images with more than 20,000 categories, has been developed and led to enhancing the classification algorithms' performances. It evolved into the annual competition named ImageNet Large-Scale Visual Recognition Challenge (ILSVR), through which classification algorithms compete with each other in terms of the least error rate. Over the past years, many algorithms are proposed, among which are: AlexNet, ZFNet, VGGNet, GoogLeNet, ResNet, Wide ResNet, DenseNet, and SENets [41]. These are feature extraction networks, or backbone networks, which are used by any object detector, like R-CNN, FASTER R- CNN, OR YOLO, …etc., as feature extractors. Fig. 2.5.1. shows a performance comparison chart between the known classification networks that have been trained on ImagNet classification dataset

2.5.4 Measures to assess the performance of a classification system

This section presents measures used to assess how good (accurate) the classifier is at predicting the image class label. Classifier measures include recognition rate (overall precision), recall (or sensitivity), specificity, precision, and F-measure. They are calculated according to equations: (2.5.8), (2.5.9), (2.5.10), (2.5.11), (2.5.12) and (2.5.13).

$$\text{Recognition rate} = \frac{TP+TN}{P+N} \quad (2.5.8)$$

$$\text{Error rate} = \frac{FP+FN}{P+N} \quad (2.5.9)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2.5.10)$$

$$\text{Specifity} = \frac{TN}{N} \quad (2.5.11)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2.5.12)$$

$$F - measure = \frac{2*precision*recall}{precision+recall} \quad (2.5.13)$$

Where

- P is the number of positive images and N is the number of negative images
- True positives (TP): These are the correctly predicted positive values, which means that the value of the real class is yes and the value of the predicted class is also yes.
- False positives (FP): when the real class is no and the predicted class is yes, i.e. these are the negative values that have been incorrectly labeled as positive.
- False negatives (FN): when the real class is yes, but the expected class is no, i.e. these are the positive values that have been mislabeled as negative.
- True Negatives (TN): These are the correctly predicted negative values, which means that the value of the real class is no and that the value of the predicted class is also no.
- F- measure is the harmonic mean of precision and recall.

TP and TN tell when the classifier is doing things right, while FP and FN tell us when the classifier is wrong (i.e., misclassification).

A confusion matrix [42] is a useful tool for evaluating the performance of a classifier in recognizing and predicting images of different classes. It is a table of size (n x n), where n is the number of classes. Each column of the matrix represents the expected classifications and each row represents defined classifications. Entries occupying the diagonal of the confusion matrix indicate a good classification with the greatest precision and entries outside the diagonal indicate classification errors.

CHAPTER 3

Related work

The purpose of this chapter is to present the works related to our work. Namely vehicle type classification. We discussed solutions that identify vehicle properties from two perspectives: and fine-grained classification, i.e. vehicle make and model, and coarse-grained classification i.e. vehicle type, we presented the state of the art in this regard in two sections. In the first section, the problem of discriminating between two very similar vehicles has been tackled under the problem type of “fine-grained vehicle classification”, where there are small variations of class-specific features distributed between different class variances. In the second section, several vehicle type classification methods have been presented. Their main focus was to classify objects initially classified into two classes; vehicle class or non-vehicle class, and then classify vehicles according to their types i.e. Crossover, Sedan, Hatchback, pickup, Van, and Minivan, etc. Finally, we presented how we benefited, in our work, from these algorithms, and where to position our work when compared with the state of the art.

3.1. Vehicle fine-grained classification

In the literature, the problem of discriminating between two very similar objects has been tackled from different perspectives. The common one is implied in the problems that have been tackled under the problem type of “fine-grained classification”, where there are small variations of class-specific features distributed between different class variances. Such problems have usually been solved by building a feature vector that includes discriminant parts features in the input images and using it for classifying the input. In [43], a solution was proposed to tackle the unsolved challenge of identifying vehicles of highly similar appearances based on the small differences between them. Authors have indicated that many applications require a fine recognition of objects’ identity which means, in the case of vehicles, make and model recognition. Several challenges and problems are

confronting the fine-grained recognition of cars. There are problems related to data acquisition and annotation and extensive variations between vehicles of different types. Additionally, which is the most important and very challenging, to identify highly similar vehicles of the same brand. The problem tackled in this work was that of data acquisition and tagging. They perform data acquisition and annotation automatically based on an object detection model through which vehicle images are annotated efficiently. A Faster-RCNN based model accompanied with an intensive dense attention network (DA-Net) is used to mitigate the impact of high information attenuation caused by the transfer of feature information between layers which have many obstructs in the traditional DNN. The solution also mitigates the lack of mutual dependency between the cascaded features layers and the poor focus on the regions/characteristics of interest. The proposed method achieved mAP of 94.5% in the Stanford Cars data set and 95.8% in the FZU Cars dataset.

Another method was proposed in [44] for recognizing the car Make and Model. It is required that the car class is precisely recognized and differentiated from other similar cars in the same category. Since there are only minor variations between these “within-category” cars, it is a real challenge to capture reliable variations, sufficient to get decisions on the cars' fine-class recognition. A novel two-step approach was proposed in which, in the first step, discriminative parts, that are agreement with human-annotated parts in the image database, are extracted visually from images under query and scored. As, since objects in the parent category share common descriptors, like shape and specific parts, the objects in sub-categories could be identified accordingly. In the second step, those discriminative parts descriptors, extracted in the 1st step, are described and given scores. Afterward, a final matching score to the image database is found based on the weighted sum of these two scores. Thus, the relevant variations among cars, which are very small and dominated by the overall characteristics of the category are captured in the proposed solution. Sedan cars in a model that have single view images as input are used to verify the solution. The method shows a classification precision of 62.53% and a classification recall of 79.09% compared to manual classification. In the same line, a coarse-to-fine CNN model for car model recognition was proposed, where there is a need to detect discriminative parts of the most significant appearance variation. In this method, the most discriminative parts’ variations are detected automatically via the CNN feature maps. The method is capable of learning the most important variations found in subordinate levels. Additionally, using the CNN multi-layered structure makes it easy to learn hierarchical features in a large dataset. The adopted feature learning process, according to the research title, is a coarse-to-fine one, which means that it detects and extracts more and more fine parts “variations” parts from global as well as local regions. Accordingly, results of the detecting procedure are forwarded further as an input to get more

discriminative regions that, by their turn, are forwarded again as input and so on. This will continue until there are no more subtle regions. In this procedure, global features reflect holistic variations while local features describe variation in subordinate levels. A combination of the global and the local features is used to train a one-versus-all linear SVM classifier. The proposed method achieved an accuracy of 98.29% with 281 vehicle makes and models and the results are evaluated using the CompuCar dataset [45].

In another coarse to fine approach for vehicle recognition in intelligent transportation systems, an approach that builds on the many successes of Convolutional Neural networks in image classification solutions was proposed. In that approach, the problem was looked at as consecutive connected processes regardless of their differences. For instance, the vehicle's location is identified from the first glance and defined by a bounding box, then the maker and shape are captured as accurately as possible, and finally, more details are captured from more observations. In that work, to generate an accurate regional proposal, the relationship between the original image and its CNN-based feature map is investigated to come up with an aware- map with high-quality proposals but their number is limited. Additionally, a new learning technique was proposed. It is a hierarchical one with multiple softmax regression by which the vehicle classification accuracy was improved. The framework uses 3 networks of different tasks to tackle the different aspects of the vehicle and to make the final decision based on all of them. The method was verified on the Stanford cars dataset and showed that it is more advantageous than the state-of-the-art methods concerned with vehicle fine-grained classification [46].

3.2 Vehicle type classification

In [47], a Transfer Learning-based vehicle classification system, by using a Convolutional Neural Network (CNN) pre-trained on a large-scale dataset, was proposed. In transfer learning, a pre-trained classification network is used as a baseline. In this regard, this pre-trained network is re-trained on a small dataset consisting of hundreds of labeled vehicle images belonging to different classes and benefits from what allowed it to reach its optimal configuration. This re-trained model is then capable of making predictions on new data and defining the estimated accuracy. In the proposed model, the system is divided into two stages; the vehicle area is detected, by Haar-like features on the roadway video, then GoogLeNet was used as a Transfer Learning-based model for vehicle classification. They reached higher accuracy than the non-transfer learning-based methods.

In [48], authors highlighted how vehicle classification was a popular field of research for many years. They also highlighted that the model, manufacturing date and manufacturer, logo,

types, and dimensions are usually the basis of vehicle classification solutions and that vehicles datasets are mostly available online for the public. However, they also indicated that using the number of axles as a basis for deep learning-based vehicle classification solutions has not yet been used, and, as such, there is no available public dataset that includes vehicle axles. Thus, a solution to perform vehicle classification based on type and number of axles was followed by a 2nd phase to categorize vehicles into five groups relevant to the Indonesia regulation for toll road tariff. ResNet CNN and Transfer Learning have been used to fine-tune the model to enhance classification accuracy. The model achieved 99% accuracy.

A real-time-based vehicle classification method was proposed in [49]. In their proposal, appearance-based vehicle classification methods were indicated. These methods are based on appearance features like front views, Sobel edges, or side views. In real applications, many front views vehicle images are captured by traffic surveillance cameras and used as a base of vehicle type classification models. In the proposed solution, front views vehicle images are used in that supervised-based solution, where a CNN is used to capture vehicles features in the training session. Additionally, an unsupervised pre-training procedure was provided due to its great usefulness in the training of CNNs. In general terms, convolution applies filters that extract useful information from the images. Convolution applies different filters to the image and obtains, as an output, a new (usually smaller) transformed image for each filter, these transformed images are commonly known as feature maps. A sparse Laplacian filter was used to capture vehicles' information and the probability of each vehicle type has been got by using a trained softmax layer. The used CNN has two stages through which low-level local and high-level global features are generated respectively. The used network is characterized by being a layer skipping one by which the two stages outputs are fed into the classifier. The authors said that their method performs well in complex scenes. They claimed that they achieved a day accuracy of 96.1% and a night accuracy of 89.6% when the model was tested on the BIT-Vehicle dataset.

In [50], authors pointed at the special interest Deep CNN has got in recent years, due to the great improvement of Neural Network-based algorithms and commented that it has become an obvious choice for many scholars who see that learning what constitutes the human visual system gradually became a huge source of inspiration for them. Besides, they indicated that the speed of GPUs had increased significantly, which made it possible for convolutional to work without a layer-by-layer algorithm. With the increase in computing speed, it became evident that Deep Learning had considerable advantages in terms of efficiency and speed. In this context, the problem addressed was that CNN-based classification systems require a huge amount of data sufficient to attain high training performance and ensure generalization. Authors have addressed that there is no

available benchmark dataset to be used in developing and testing vehicle type classification systems. They claimed that the available vehicle datasets, like CompCr and Stanford cars, are regionally based, i.e. give satisfactory results when they are used in Intelligent transportation systems in specific regions, but their performance has got criticized when there are non-regional vehicle classes. To tackle these issues, they proposed a CNN-based system in which six-vehicle classes were categorized in adverse illuminating conditions. A performance comparison, in terms of accuracy and convergence, was conducted between 6 pre-trained CNNs who are fine-tuned and tested. The comparison concluded that ResNet was the optimum model. It was improved further by adding a new classification block to it and thus achieved bitter performance. On a custom dataset, they claimed that their method achieved 99.68%, 99.65%, and 99.56% accuracy, precision, and F1-score respectively.

In [51], It was pointed at how the CNN-based image classification large scale training dataset is becoming essential to learn features of each category to the level that facilitates obtaining satisfactory classification results in real implementation tasks. However, the constraint of the need of having a huge amount of training data to automatically adjust the CNN sophisticated parameters represents a real obstacle in using CNN- based classification methods. Considering that constraint, a system was proposed through which vehicle classification is conducted in two phases. GoogLeNet is used in the first pre-training phase to get the initial model with its appropriate connection weights. It was pre-trained on the ILSVRC-2012 dataset. Thus, by this step, extracting features manually was avoided. In the second phase, the model is fine-tuned on a dataset that has 13,700 vehicles distributed on six categories taken from real highway surveillance videos. The average classification accuracy reached 98.62%.

In [52], authors have highlighted how analyzing and extracting information from traffic surveillance cameras is extremely important in modern traffic systems. Accordingly, they proposed an MIO-TCD dataset-based vehicle classification and localization method. This dataset has been recently released and is representative of the real traffic surveillance environments. They used the available feature for stacking many layers in ResNet, which is easy to be used, as a basis of their solution. In this method, the authors used a technique called Joint Fine-tuning (JF) to improve the classification performance and a dropping CNN (DropCNN) method as a synergy to the JF. In performing the localization task, they used a region-based detector combined with a feature extraction network constructed from ResNet50 and 101 layers. They claimed the accuracy rate is 97.95%, which is the highest among several state-of-the-art methods.

From another viewpoint, authors in [53] indicated that detecting objects correctly from images and videos are the first step in any visual surveillance system after which they are classified

to their different vehicle category types e.g. cars, buses, trucks, etc. They highlighted that many vehicle type classification methods don't consider the existence of minority categories and, as such, those categories are misclassified because the high percentage of their correct predictions is looked at as an indication of being dominant. Thus, this imbalance of classes induces a decrease in Neural Network learning performance and raises a need for a solution that handles the under-represented classes categories. Thus, their objective was to confront the issue of imbalanced data collected from traffic surveillance in vehicle classification methods. A method was proposed for classifying imbalanced data by integrating CNN with balanced sampling. The method has two phases. In the first phase, the unbalanced dataset problem is mitigated by applying data augmentation with balanced sampling. In the second, they used the augmented training dataset to construct an ensemble of CNN's different architecture models with new parameters. They claimed that the overall accuracy of all categories has been enhanced as compared with the baseline algorithms.

In the same line, authors in [54] tackled the misclassification of images belonging to rare categories as compared with majority classes. To avoid that it is not advised to assume that all samples have equal error costs. Thus, it is suggested that if we got high overall accuracy we enhance the mean accuracy of all categories. Additionally, deep neural networks are used extensively in the last decade in machine learning tasks. One of the main problems in using these networks is that this quantity is not always available (data missing, not accessible, or too expensive). The artificial increase in data was introduced to resolve this problem and has become one of the best practices for improving the performance of CNNs. There are several approaches to data augmentation, among which is Generative adversarial networks (GANs) which have been used to generate synthetic data from the existing database itself with similar characteristics to that database. However, this approach may not achieve satisfactory performance because of the gap between the distribution of the synthetic images and that of the real ones for rare classes. The authors proposed a method based on Generative Adversarial Nets (GAN) through which they integrated DNNs with data augmentation. In the first, they generated adversarial samples for the rare classes by using the original dataset to train several GANs. In the second, an ensemble of different CNN models has been trained on the original imbalanced dataset after which the low-quality adversarial samples were filtered out from them. In the end, the model has been refined by compiling the selected adversarial samples and the augmented dataset. Their method increases the classification performance of some categories while maintaining a high overall accuracy compared with the existing methods.

In the previous literature review, the Deep Learning-based vehicle type classification models are diversified. They use different methods to achieve significantly higher accuracy with

different state-of-the-art technologies. Concerning the fine-grained classification of vehicles, several methods to recognize the car Make and Model are presented. The literature shows that such problems have been solved very often by finding discriminant parts and extracting features from the input images and combining them to build the final feature vector which is then used for classifying the input. The reviewed models in [43, 44, 45, 46,] have used transfer learning, SVM, and GANG technologies.

As for vehicle type classification, which is the classification category to which our work belongs, the reviewed models in [47, 48, 50, 51, 52] have used a combination of Transfer Learning and Deep Learning while others are Deep Learning-based ones. In [50, 53, 54], they are using data augmentation techniques to artificially modify or increase the training data to improve the model performance. In [47, 48, 49, 51], they used a custom dataset for training which could be a source of performance degradation because of its limited size as compared with the real-world data.

Tab. 3.1 Summary of Vehicle classification research.

No.	Reference	Technology	Technique	Dataset
1	Fine-grained vehicle type detection and recognition based on dense attention network	Faster-RCNN, Dense Attention Network (DA-Net)	Mitigate the impact of high information attenuation caused by the transfer of feature information between layers.	Stanford cars dataset, FZU cars dataset
2	Car Make and Model recognition combining global and local cues	Proprietary classifier	Annotating discriminative parts in the input images manually and using global shape and appearance descriptors for vehicle recognition.	The dataset contains 10 different sedan types car
3	Fine-Grained Vehicle Model Recognition Using A Coarse-to-Fine Convolutional Neural Network Architecture	SVM	Detect discriminative parts of the most significant appearance variation	CompuCar dataset
4	Transfer Learning-based Vehicle Classification.	Haar algorithm, GoogLeNet, Transfer learning	Vehicle area is detected, by Haar-like features, then GoogLeNet is used for vehicle classification.	Limited scale vehicle dataset

5	Transfer Learning-based Vehicle Classification	Haar algorithm, GoogLeNet, Transfer learning	Vehicle area is detected by Haar-like features, then GoogLeNet is used for vehicle classification.	Limited scale vehicle dataset
6	Indonesia Toll Road Vehicle Classification Using Transfer Learning with Pre-trained ResNet Models	ResNet, Transfer learning	The images are classified manually based on the number of axles used to distinguish between vehicles	Dataset collected by a smartphone camera
7	Vehicle Type Classification Using a Semi-supervised CNN	Semi-supervised CNN	A sparse Laplacian filter is used to capture vehicles. A trained softmax layer is used to get the probability of each vehicle type.	BIT-Vehicle dataset.
8	Convolutional Neural Network Based Vehicle Classification in Adverse Illuminous Conditions for Intelligent Transportation Systems	ResNet, Transfer learning	Improving ResNet is used by adding a new classification block.	Custom dataset including 10,000 images of six vehicle classes
9	Vehicle Classification for Large-scale Traffic Surveillance Videos Using CNN	GoogLeNet, Transfer learning	Fine-tuning the CNN model.	Custom vehicle dataset including 13,700 images of six classes
10	ResNet-Based Vehicle Classification and Localization in Traffic Surveillance Systems	ResNet 18	A technique called joint fine-tuning (JF) and a dropping CNN (DropCNN) method as a synergy to the JF.	MIO- TCD traffic dataset
11	An Ensemble Deep Learning Method for Vehicle Type Classification on Visual Traffic Surveillance Sensors	CNN, Data augmentation	Applying data augmentation with balanced sampling and using the augmented training dataset to construct a different architecture with new parameters.	MIO- TCD traffic dataset
12	Improving Deep Ensemble Vehicle Classification by Using Selected Adversarial Samples	Generative Adversarial Nets (GAN), data augmentation	Integrated DNNs with data augmentation in 3 successive phases.	MIO- TCD traffic dataset

In these models, we expect that there is still a noticeable performance gap between lab test results and real-world results caused by a lack of efficient training data. For instance, their lab accuracies are 98.30%, 99%, 95.70%, and 95, 49% respectively, and real-world results will be lower, such as in [50] where the lab accuracy is 99.68% and the real world one is 97.66% when tested on the VeRi dataset with an accuracy gap of 2.02%. Therefore, from this analysis, we consider that there is a need for a data-driven technique that is capable of minimizing the performance gap between the lab test and the real-world results to the lowest possible level. We provide an optimum vehicle classification solution that is independent of the adopted CNN. It is based on using an effective novel augmentation technique that outperforms the current data augmentation. The technique is capable of raising the efficiency of the data in hand to make the model performs better and achieve higher accuracy when tested on real-world data. Besides, we integrated the proposed novel augmentation technique with Transfer Learning technology to maximize the acquired efficiency. Table 3.1. presents a summary of vehicle classification research.

CHAPTER 4

Proposed method

In this chapter, we mainly focus on how the proposed method is going to solve the research problem. We highlighted the concept of the pre-trained networks and focused on ResNet specifically which we adopted in our method. We also showed how ResNet based retrained model is better than the original model in terms of accuracy, processing time, etc. Additionally, we described how we used data duplication, to enhance vehicle type classification. We showed that if we duplicated the whole dataset before splitting it into training and testing sets it will lead to over-fitting because, in this case, the model is tested on data that it has already seen, somehow, in the training dataset and results in unrealistically high-performance levels in the testing dataset. We showed, in the proposed method, how duplication is performed only on the training dataset so that the testing dataset contains unique samples and, accordingly, represents the real-world data. We then described how the dataset was organized and how we split it into training, validation, and testing sets. We also highlighted the function of each set and the number of vehicles per class included in each set. The chapter ends up with describing the model training and how the proposed augmentation technique affects the training process and defines the criterion of ending the training. The flowchart of the proposed method is shown in Fig. 4.1. The implementation process comprises 3 main operations; Transfer Learning, data augmentation using data duplication technique, and generalization testing process.

4.1 Transfer learning

4.1.1 Image classification using Deep Learning

Convolutional Neural Network (CNN) is a special type of Artificial Neural Network explained earlier. It was introduced in the early 1990s [55]. CNNs have their “neurons” arranged like those of the front

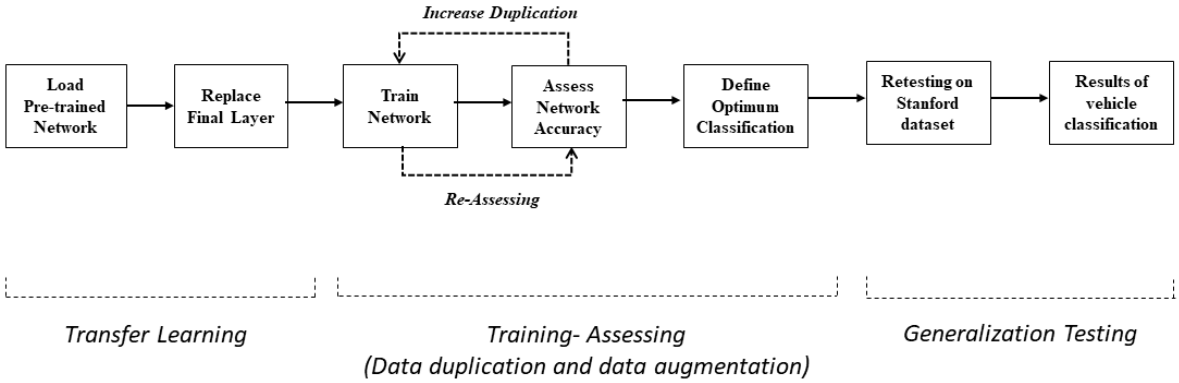

Fig. 4.1. Flowchart of the proposed method.

lobe, which is the area responsible for processing visual stimuli in humans and animals [56]. Their name reflects one of the most important operations in the network convolution [57]. Most convolutional networks used to process images, perform two operations: the first is the aforementioned convolution, and the second is an operation called pooling. In general terms, convolution applies filters that extract useful information from the images. Convolution applies different filters to the image and obtains as an output a new (usually smaller) transformed image for each filter, these transformed images are commonly known as feature maps. The ReLU activation function which stands for a rectified linear unit is applied at the output of a convolution layer. It breaks (a part of) the linearity by removing some of the values (all negative), i.e., It replaces all negative numbers with the value 0, and thus speeds up calculations as well. It does not impact the features highlighted by the convolution. Finally, all feature maps are reduced again extracting the most significant values with the pooling operation. In this way, the image decreases in size and is found simplified (smoothed). This process can be repeated in a loop repeatedly in a convolutional network. Once the desired features have been extracted, after applying several convolutions and proper pooling, the output of the last pooling layer is connected to a conventional Neural Network to classify the image. When we go to further details, we can say that a Convolutional Neural Network, according to [57], is a Neural Network that uses a mathematical linear operation called convolution or product of convolution and each Convolutional Neural Network contains the following:

a) Convolutional layer

The Convolutional network contains at least one convolution layer. Let f and g *be* two functions defined on R, the convolution product between f and g is generally denoted $f * g$ and it is defined by the following equation (4.1.1).

$$s(x) = (f * g)(x) = \int_{-\infty}^{\infty} f(t)\, g(x-t)\, dt \quad (4.1.1)$$

In a Convolutional Neural Network, the first argument *f* is assimilated to the input, the second argument *g* is assimilated to the convolution kernel and the output is assimilated to the feature card. In reality, Convolutional Neural Networks use a discrete convolution which is defined by (4.1.2):

$$s(x) = (f * g)(x) = \int_{a=-\infty}^{\infty} f(t)\, g(x-a) \quad (4.1.2)$$

In machine learning, the input is always a multidimensional data array, and the kernel is always a multidimensional parameter array that will be adopted by the learning algorithm. These arrays are referenced by tensors. Convolution is always used with a dimension greater than 1. The most used convolution is a 2D convolution. In this case, for an image input I and a kernel K, the discrete convolution is written as in (4.1.3).

$$S(I, j) = (K * I)(I, j) = \sum_m \sum_n I(i - m, j - n) K(m, n) \quad (4.1.3)$$

According to this formula, image I and the convolution kernel K are traversed in opposite directions. Indeed, when *m* increases, the indices of I decrease, and the indices of K increase. The only reason to use convolution is its commutative property. However, this property is not important in the Neural Network implementation because the other functions with which convolution is used are not commutative. It is for this reason that most network libraries of neurons implement the function of cross-correlation but they call it convolution (2.4.4). This operator is identical to convolution except that it loses the property of commutatively:

$$S(I, j) = (K * I)(I, j) = \sum_m \sum_n I(i + m, j + n) K(m, n) \quad (4.1.4)$$

A convolutional layer is characterized by:

- The dimensions of the convolution kernel (rows and columns), which must be smaller than the input image dimensions.
- The number of convolution filters C, is the number of feature cards, or feature maps, at the output of the layer. These cards are represented under the form of tensors of the 3 dimensions (H, W, C) with H the height of the maps, W the width, and C the number of channels.

- The convolution step, or stride, *s*. It is the shift step of the convolution kernel at each calculation. For example in the case of two dimensions $s = (s_1,s_2) = (1, 1)$, the kernel convolution will be moved one pixel to the right for each core horizontal movement and one pixel down for each vertical movement of the kernel when creating the feature map.
- The padding *p*. It is the parameter allowing to exceed the size of the image by adding pixels around the image.

In [58], A convolution layer at the start of a neuron network was introduced to extract images' features in a relevant way through the convolutions' nucleus. Note that the relevance of features is proportional to the late intervention of convolutions. That is, the more the convolution occurs later, the more complex the nuclei become and the more its capability of detecting more details. i.e. The deeper the network, the more details we get. Indeed, at the first convolution layer, the values of a kernel are initialized randomly. Then these values will be updated with the network learning process to improve feature extraction results.

Convolutional layers are usually followed by a ReLU layer (introduced earlier) to modify the output feature maps and called, accordingly, rectified feature maps. Applying the ReLU function to the output of the convolution layers has several advantages:

1. convolution performs addition and multiplication operations, which maintains the linearity of the output with respect to the input. Applying the ReLU function introduces a certain non-linearity (by removing pixels of negative value).
2. following the deletion of part of the data, the ReLU function allows the acceleration of calculations.
3. Enhancement of the features extracted by the convolution layers through accentuating the gap between them (negative values).

b) Pooling layer

Similar to the convolution layer, the pooling layer is responsible for reducing the spatial size of feature maps, but it retains the most important information. There are different types of pooling including Max Pooling, Average Pooling, etc. Pooling consists of applying a kernel of size n × n on the activation card by dragging it with a previously defined pitch (the pitch is generally equal to the size of the kernel n to avoid the phenomenon of overlap). Max Pooling returns the maximum value of the part of the image covered by the kernel. Instead of taking the maximum, we could take the average of all the elements covered by the kernel, this is ensured by Average Pooling. Max Pooling eliminates noise. On the other hand, Average

Pooling simply performs dimensionality reduction as a noise suppression mechanism. Therefore, in practice, Max Pooling is used much more than Average Pooling since it works better. The most common form is a pooling layer with 2 × 2 size applied in a step of 2 thus reducing the size of the input by 2, eliminating 75% of the activations, and leaving the depth dimension unchanged. In Fig. 4.1.1., the number of activations goes from 16 to 4 by applying pooling. More generally, the pooling layer:

- accepts a tensor of size $W_1 \times H_1 \times D_1$
- requires two hyperparameters: size F and step s with $F < W_1$ and $F < H_1$
- produces a tensor of size $W_2 \times H_2 \times D_2$ where
  - $W_2 = \frac{(W_1 - F)}{s} + 1$
  - $H_2 = \frac{(H_1 - F)}{s} + 1$
  - $D_2 = D_1$
- Enter zero parameters since it calculates a fixed function of the input.

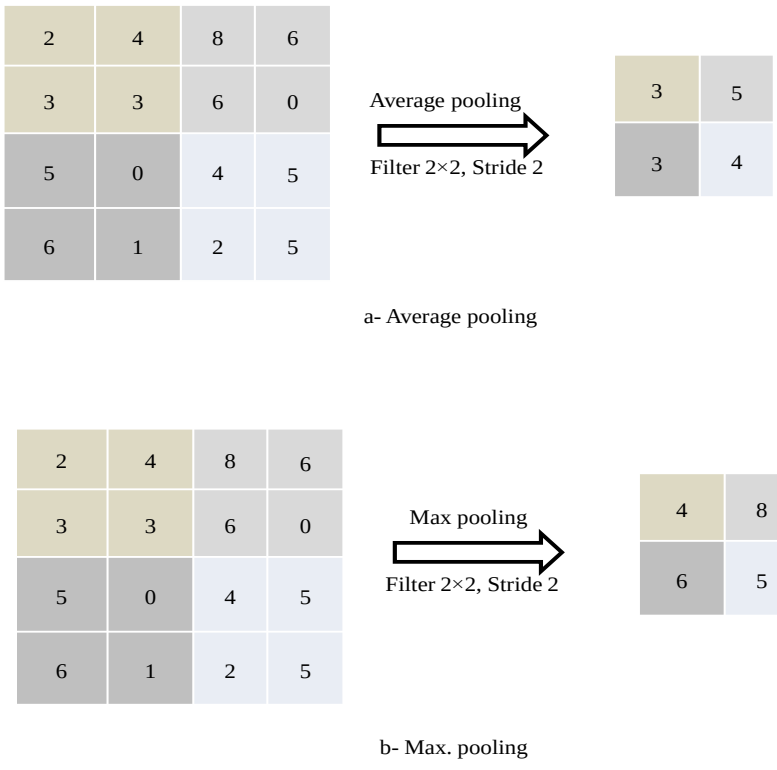

Fig. 4.1.1. Examples of average pooling and Max pooling.

- The pooling function gradually reduces the spatial size of the input data. Indeed, it allows to:

- Make input representations (feature map) smaller and easier to manage.
- Reduce the computing power required to process the data by reducing the dimensions and therefore it controls over-learning.
- Extract the dominant features which are invariant in rotation and position, thus maintaining the efficiency of the learning process of the model.
- Help to obtain an equivariant representation of the image. This is very powerful since it can detect objects in an image, no matter where they are located.

c) Fully connected layer

The fully connected layer is a traditional multi-layered perceptron that uses an activation function (e.g. softmax) on the output vector to add non-linearity. The term "fully connected" implies that every neuron in the previous layer is connected to every neuron in the next layer. Their activations can therefore be calculated with a matrix multiplication followed by a bias offset. The output of the convolution and pooling layers represent high-level features of the input image. The goal of the fully connected layer is to use these features to classify the network input image into different classes based on the learning database. In addition to classification, adding a fully connected layer is generally an inexpensive way to teach non-linear combinations of features. Most of the feature maps of convolutional layers and pooling ones can be useful for the classification task, but a combination of these features can be even better. Fully connected layers are usually followed by a Dropout [59]. The latter acts on the weights of these layers to deactivate a certain number of neurons to reduce the number of parameters. This makes it possible to control the over-learning which can be caused by a large number of parameters. The sum of the probabilities at the output of the fully connected layer is $\sum_i \hat{y} = 1$. This is guaranteed by using Softmax as an activation function in the output layer of the fully connected layer. The Softmax function takes a vector of arbitrary real-valued scores and reduces it to a vector of values between zero and one which is equal to one. It is written in the form (4.1.5).

$$\hat{y}_i = \text{softmax } (x)_i = \frac{e^{x_i}}{\sum_j e^{x_i}} \tag{4.1.5}$$

CNN-based models require millions of parameters to train on, which limits their implementation in many practical cases. In this implementation, the first layer of a CNN will learn small details of the image; subsequent layers will combine the previous features to create more complex information. In CNN, the feature extraction is performed by using filters. The

network applies filters to the image to see if there is a match i.e. the item's shape is the same as part of the image. The large size and quality of the samples used in training allow the generation of robust Deep Learning models. Pre-trained models, that mitigate the impact of the shortage of training samples, have been used by many researchers in different domains. In our work, ResNet- 50 was used in the context of transfer learning for vehicle type classification. Fig. 4.1.2 shows the configuration of ResNet-50 in Transfer Learning. The network performs the initial convolution and max-pooling using 7 × 7 followed by 3 × 3 kernels in the first two layers. There are 4 consecutive convolution blocks with 3, 4, 6, 3 layers stacked one over the other, respectively. For example, the configuration of the first layer in the first block is 1 × 1, 64 means that the convolution operation is performed by a kernel size 64 with a 1 × 1 convolution filter.

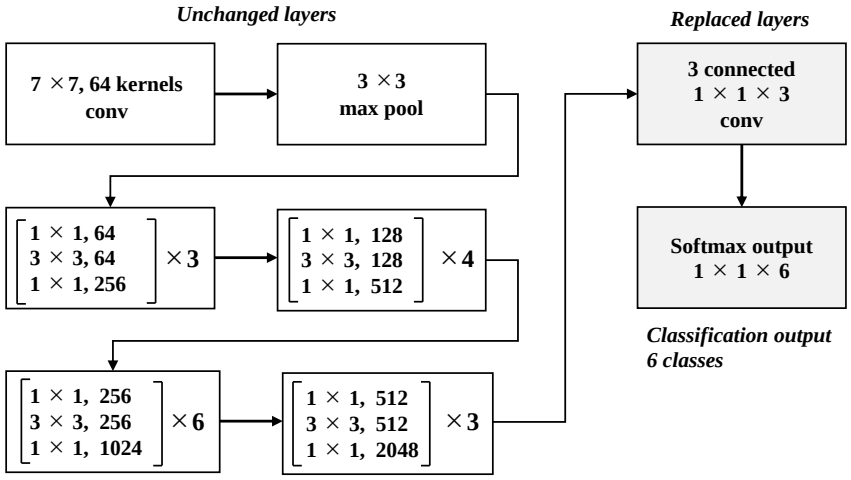

Fig. 4.1.2. Architecture of ResNet-50 in transfer learning for obtaining 6 classes output.

4.1.2 Transfer learning in the context of deep learning

Transfer learning is called so when it is applied in the machine learning context. But when it is applied in the Deep Learning context, it is then called Deep Transfer Learning. This is what we implement in our work. The Transfer Learning task, in the machine learning context, has been defined earlier by [DS, TS, DT, TT, $f_T(.)$]. Thus it is a deep Transfer Learning if the $f_T(.)$ is a nonlinear function reflecting a Deep Neural Network. According to [60], deep Transfer Learning is classified into 4 categories:

1- Instances-Based DTL

In instances-based DTL, a specific weight adjustment strategy is used. Selected instances from the source domain are appropriately reweighted and used as supplements to the target

domain dataset. i.e Instances in the source domain which are dissimilar from the target domain are excluded from the training dataset; while instances in the source domain which are similar to the target domain are included in the training dataset after being appropriately reweighted.

2- Mapping-Based Deep Transfer Learning

In Mapping-based DTL, instances from the source domain and target domain are simultaneously mapped into a new data space where they all become similar and suitable for a unified DNN. All instances in the new data space are considered as the training set of the DNN.

3- Network-Based Deep Transfer Learning

In Network-based DTL, a partial network, that was pre-trained in the source domain, is transferred to be reused as a part of the DNN in the target domain with its all network structure and connection parameters. A representation of the Network-Based DTL is shown in Fig. 4.1.3. In that figure, the network was trained first in the source domain with a large-scale training dataset and then part of that network is transferred to be part of the DNL of the target domain. This transferred sub-network might be updated later in a fine-tuning strategy.

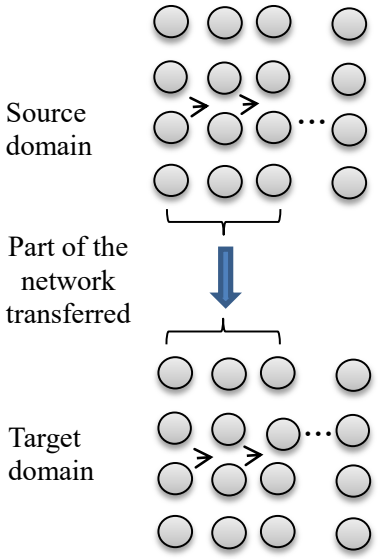

Fig. 4.1.3. Representation of the Network-Based DTL.

4- Adversarial-Based Deep Transfer Learning

In Adversarial-based DTL, adversarial technology inspired by generative adversarial nets (GAN) is introduced to find transferable applicable representations that are viable to both the source and the target domains. In that model, the training process is performed first in the

source domain (on the large-scale dataset), the network front layers are regarded as a feature extractor where features are extracted from two domains and sent to the adversarial layer. The task of the adversarial layer is to discriminate features' origin. If the performance achieved by the adversarial network becomes worse, it means that there is a small difference between the two features' types and that the transferability is bitter, and vice versa. In the afterward training process, the adversarial layer will force the transfer network to discover more transferable general features.

In our work, we used the "network-based transfer learning" mentioned earlier. The network is ResNet - 50, which is used as a baseline pre-trained classification network. In this regard, this pre-trained network is re-trained on a small dataset consisting of hundreds of labeled vehicle images belonging to different classes and benefits from what allowed it to reach its optimal configuration. This re-trained model is then capable of making predictions on new data and defining the estimated accuracy. Fig. 4.1.4 shows the Transfer Learning workflow where late layers of the pre-trained network ResNet- 50 are removed to make the model specialized only on the new training dataset on which the model has been re-trained.

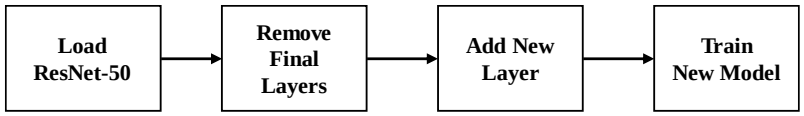

Fig. 4.1.4 Flowchart of the transfer learning process in ResNet-50.

In the implementation of Transfer Learning, the pre-trained network, ResNet-50, which is pre-trained by 1 million images of 1000 classes, is loaded. Then, its fully connected layer and classification layer are replaced with new ones adapted to the new dataset. These replaced layers contain the required information needed to combine the extracted features into class probabilities that consist of a loss value and predicted labels. The new fully connected layers have several outputs equal to 6 classes.

4.2 Training assessing

We describe the dataset structure and how it was prepared. The recurring cycle of training- assessing is presented and both augmentation and duplication techniques are described.

4.2.1 Prepare dataset

The custom dataset is composed of six classes from the Stanford cars dataset [61] . Fig. 4.2.1 shows an

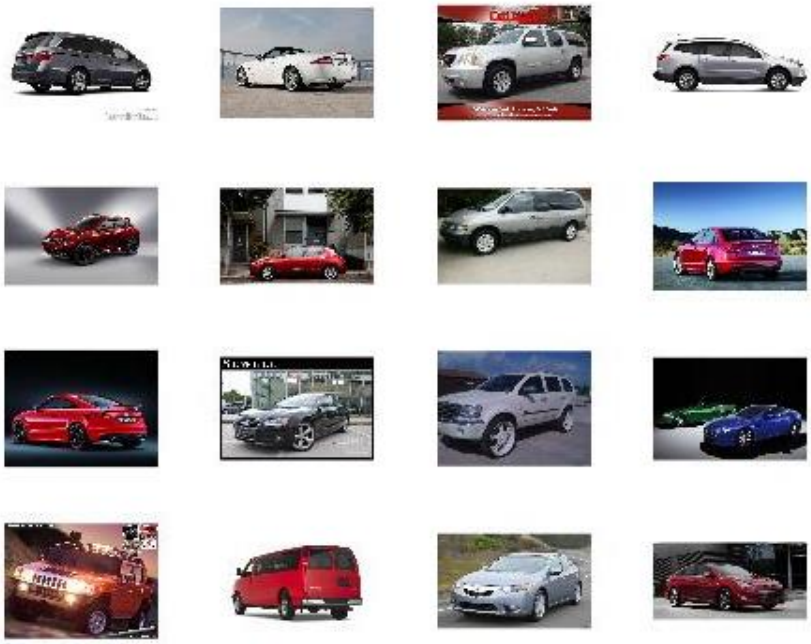

Fig. 4.2.1 Example of vehicle images from the training dataset.

example of vehicle images from the dataset. The dataset includes 2128 car images, grouped into six classes manually, Crossover, Sedan, Hatchback, Van, Pickup, and Minivan. In the procedure, the dataset is divided into the training set 70% and the test set 30% as shown in Fig. 4.2.2. The validation set is formed by a random selection of 10% in the training dataset.

The training dataset is used to learn the model vehicle types. The test dataset is used to verify that the model is capable of classifying the dataset. Table 4.2.1. shows the structure of the dataset after being classified into six categories. Each class has, approximately, the same number of images as the following, Crossover 362 images, Sedan 351 images, Hatchback 339 images, Van 352 images, Pickup 368 images, Minivan 356 images, respectively. The validation set is formed by a random selection of 10% of the images in the training dataset and contains (20) images in each class. In the testing set, each class has approximately the same number of images as the following, Crossover 108 images, Sedan 106 images, Hatchback 102 images, Van 106 images, Pickup 111 images, Minivan 107 images, respectively, with a total number of 640 images. The training dataset instances are of different sizes, while the network requires that input image size is 224 × 224 × 3. Therefore, the input images are resized to this resolution.

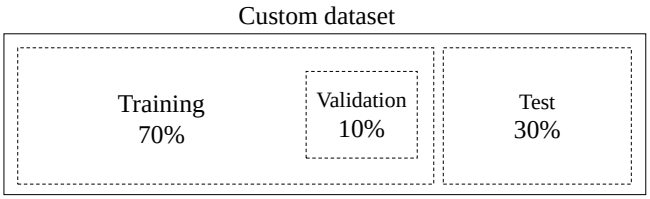

Fig. 4.2.2. Structure of the custom dataset, Training set 70% including Validation set 10% and Test set 30%.

In addition, two data augmentation techniques are performed on those images before input to the training stages. In the first, training set instances are duplicated in a staged procedure. Then, images are randomly flipped along the vertical axis and translated horizontally and vertically up to 30 pixels in a random way.

Tab. 4.2.1. The dataset after being classified into six classes.

Class / Dataset	Crossover	Sedan	Hatchback	Van	Pickup	Minivan	Subtotal
Training	254	245	237	246	257	249	1488
Validation	(20)	(20)	(20)	(20)	(20)	(20)	(20)
Testing	108	106	102	106	111	107	640
Total	**362**	**351**	**339**	**352**	**368**	**356**	**2128**

4.2.2 Recurring cycle of training assessing

The model training is performed in a staged manner. In the first stage, the training dataset has only the baseline vehicle instances in the dataset without any duplication. In every subsequent stage, the number of duplications per image is increased by one and the training is repeated. The model is evaluated afterward on the test dataset where the overall accuracy is recorded. This cycle of multiple activities (i.e. duplication, training, and assessing) is repeated several times, with the number of data duplications increased by 1 each time, until the maximum classification accuracy is reached.

4.2.3 Data augmentation

As mentioned earlier, to address the lack of learning data, several methods for artificially producing new examples of data were used. The effect of increasing data is simply because it gives redundancy that helps to learn. It is then possible to carry out training on sets containing real and synthetic data in

controlled proportions. The objective is to extend the training databases without not only losing representativeness but also respecting this constraint. The notion of control (mastery and knowledge) of the transformations applied is also important for a better understanding of the effects produced by the final system after learning. For instance, in the context of vehicle classification, CNNs have shown remarkable precision and competitive reliability compared to traditional methods. However, according to [62], CNN-based approaches require that the network be driven over a large amount of data. One of the main problems is that this quantity is not always available (data missing, not accessible, or too expensive). The artificial increase in data was introduced to resolve this problem and has become one of the best practices for improving the performance of CNNs. There are several approaches to geometric transformation-based data augmentation presented in section 2.5, among which are flipping and translation [63] which we used in our work. These techniques allow the CNN model to learn more diverse image features and therefore be able to correctly predict the category of the captured image.

1- Flipping

Flipping is a technique in which an inversion is done on the original image. This inversion can be horizontally or vertically. Reversing the horizontal axis is much more common than inverting the vertical axis. This increase is one of the easiest to implement and has proven to be useful in datasets like CIFAR-10 and ImageNet.

Fig. 4.2.3. Horizontal flip.

In datasets that involve text recognition such as MNIST or SVHN, this is not a label preservation transformation. For example in the MNIST database, where it includes the identification of numbers from 0 (zero) to 9 (nine), the number 6 (six) could be confused with the number 9, after performing a horizontal and then a vertical inversion. Fig. 4.2.3. shows an example of a mirror image.

2- Translation

Changing the images left, right, up, or down can be a very useful transformation technique to

avoid positional distortions in the data. For example, if all images in a dataset are centered, which is common in face recognition datasets, this would require the model to be also tested on perfectly centered images. When the original image is translated into one direction, the remaining space can be filled with a constant value, such as 0 or 255, or it can be filled with random or Gaussian noise. This fill preserves the spatial dimensions of the post-magnification image. This enhancement method is very useful as most objects can be located almost anywhere in the image. This forces the Convolutional Neural Network to search everywhere in the image as an example is shown in Fig. 4.2.4.

(a) Original image (b) Right translation (c) Right and down translation

Fig. 4.2.4. Translation example.

In the proposed method, these two augmentation techniques are implemented to increase the size of the training dataset as shown in Fig. 4.2.5. The training dataset is duplicated first, then these basic augmentation techniques are implemented where the training images are randomly flipped along the vertical axis and randomly translated up to 30 pixels horizontally and vertically. Therefore, the size of the dataset is artificially increased in the learning process as the data augmentation created more image variants in the training dataset. The created image variations improve the ability of the model undergoing training to be generalized and highly effective. We proved empirically that augmenting data by duplication enhances classifier performance.

The training process is conducted in several stages. At each stage, the number of data duplications is increased, and also the model is re-assessed as shown in Fig. 4.1. The loop of increasing data duplications and re-assessing is continued until the optimum performance is reached. The learning process continues in the context of Transfer Learning until the model reaches the desired stop criterion which is the maximum number of data duplications in the training dataset.

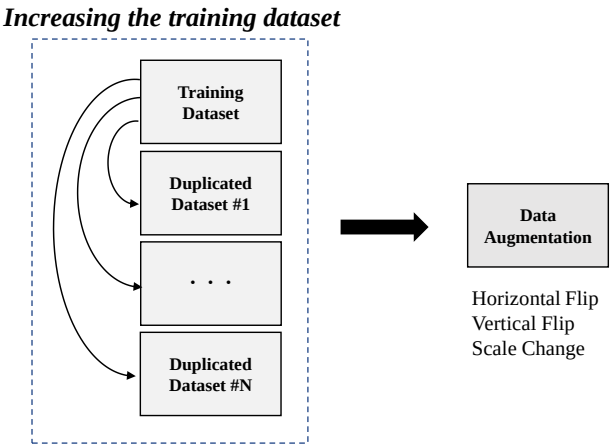

Fig. 4.2.5. The proposed data augmentation process using data duplication for increasing the size of training dataset.

4.2.4 Data duplication.

Data duplication in a dataset is generally avoided in Deep Learning because when dividing a dataset into testing and training sets, the goal then is to ensure that no data is shared between the two sets. This is because the purpose of the testing process is to simulate real-world data. Thus, when there are duplicates in the original dataset, the model is tested on data that has already been seen in the training dataset and results in unrealistically high accuracy. The proposed method used data duplication as one of the data augmentation techniques as shown in Fig. 4.2.5. The data duplication is performed only in the training dataset so that the testing dataset remains to have unique samples and thus represent the real-world data. Accordingly, the data duplication increases the size of the training dataset artificially and thus the learning process is improved since there are more image variants on which the DNN model is trained. The training is performed in consecutive stages. In each stage, the model is trained on the duplicated training dataset and evaluated on a testing dataset that has no duplicates. In this context, the baseline instances, as well as the duplicated ones, are exposed to the applied simple augmentation techniques which are: flipping the training images randomly along the vertical axis and translating them up to 30 pixels horizontally and vertically in a random way. Thus, during the training process, each instance in the dataset appears with different variations as many times as its duplicate counts in each epoch. By the proposed data augmentation, it is ensured that the model will not be over-fitted due to the limited training data as, according to [64], the limited training data might make the model unable to generalize. i.e. It is a good model when classification is based on the features included in the small training dataset, however, it won't be able to do good classification for other objects with features that it has not been trained on.

4.3 Generalization Testing

In Deep Learning, if the model can adapt to the training data, it does not mean that it will perform well on the test data that represents the real world. This disparity between performance on training and real-world data is called the generalization gap. In the proposed method, to generalize and confirm the achieved results, generalization testing is implemented when the model reaches the desired stop criterion which is the maximum number of data duplication in the training dataset which leads to the highest accuracy level. The ResNet-50 based proposed model was then tested on the generalization test set which is a Stanford based dataset consisting of 8,000 images grouped into 6 categories. The objective is to prove that the network not only behaves properly with the testing dataset but also behaves similarly with a real-world dataset.

CHAPTER 5

Experiments

In this chapter, we described the testing environment, the model training, and the staged training. We also presented the training parameters and indicated every type of training. From the experimental results, the optimum classification is identified. We also described how staged training was performed. We highlighted the baseline training, in which the basic data augmentation was implemented, and depicted the consecutive training sessions that follow the baseline one.

We highlighted the un-changeability of the training parameters while we increased the number of duplications one by one in every subsequent stage. We also showed how the optimum classification is identified in the experimental results. Furthermore, to ensure the model generalization, we highlighted testing the model on the Stanford image dataset which represents the real-world data. Then, there is a discussion about the experimental results where we showed how they fulfilled the work goal and how the learning process was improved and the classification performance was enhanced by performing the proposed augmentation technique based on data duplication. At the end of this chapter, a comparison between the performance of the proposed method some state-of-the-art vehicle classification methods was depicted.

5-1 Model training

5.1.1 Environment

The proposed method was implemented on the MATLAB environment and verified using a GPU-enabled PC with the following specification: NVIDIA GeForce RTX 2080 Ti, Processor Intel Core i7-9700 CPU@300 GHz, 3000 MHz, 8 Cores, and 8 logical processors

5.1.2 Perform training

We performed a staged training where we started with a baseline training in which the basic data augmentation was implemented, and followed it with consecutive training stages. In the staged training, the proposed data augmentation technique was performed which the training dataset instances are duplicated in the subsequent stages as 0, 2, 3, 4, and 5. The classification performance in each stage was recorded and the training progress in each one was also indicated.

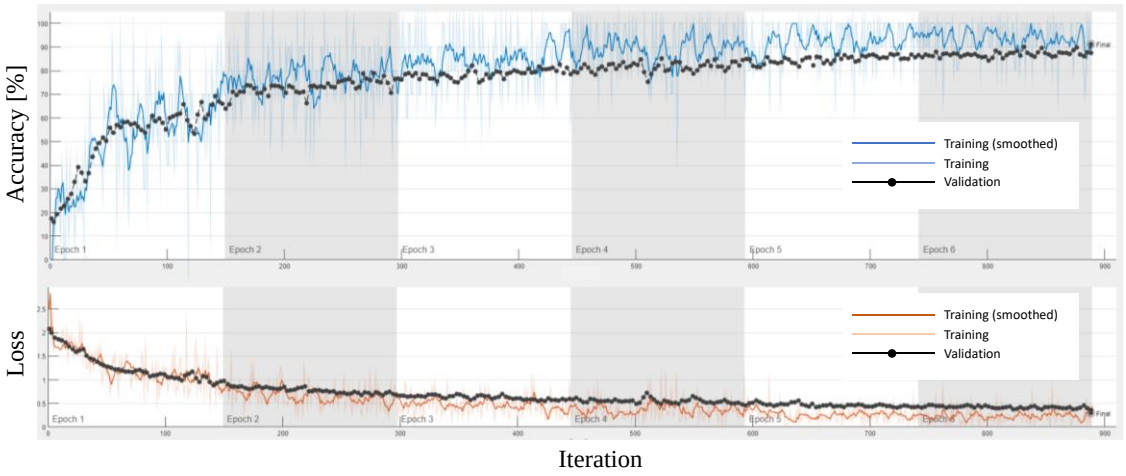

Fig. 5.1.1. The progress of the baseline training at the stage 0 in the proposed method with 6 epoch, learning rate 0.0001, batch size 10.

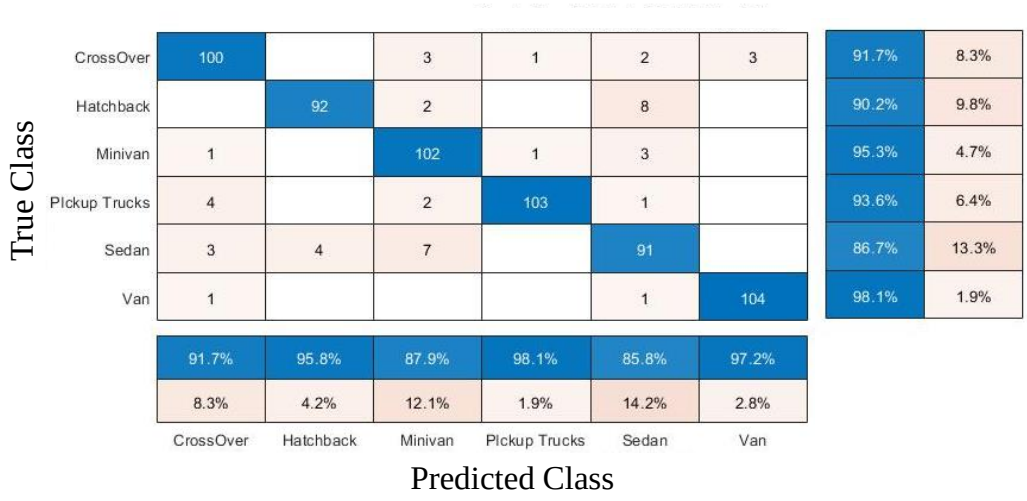

Fig. 5.1.2. The confusion matrix of validation dataset in the baseline stage 0.

a- Baseline training. Training parameters were set as the following; the number of epochs 6, learning rate 0.0001, and batch size 10. Fig. 5.1.1. shows the progress of both training and validation of the baseline model training. It is noted that the over-fitting was not seen during the training progress. In the baseline training, we implemented the general data augmentation on the training dataset before commencing the training and tested the model afterward on the testing dataset. Fig. 5.1.2. shows the confusion matrix of the testing dataset classification results after the baseline training was completed.

b- Staged training. The training parameters were unchanged while increasing the number of duplications one by one in every subsequent stage. When the training was finished, the model has evaluated afterward on the testing dataset and the overall classification accuracy was recorded.

Tab. 5.1.1 Configuration of training dataset with data duplication in each stage.

class / Stages	Crossover	Sedan	Hatchback	Van	Pickup	Minivan	**Total**
0 (original)	254	245	237	246	257	249	**1488**
2	508	490	474	492	514	498	**2976**
3	762	735	711	738	717	747	**4437**
4	1016	980	948	984	1028	996	**5952**
5	1270	1225	1185	1230	1285	1245	**7440**

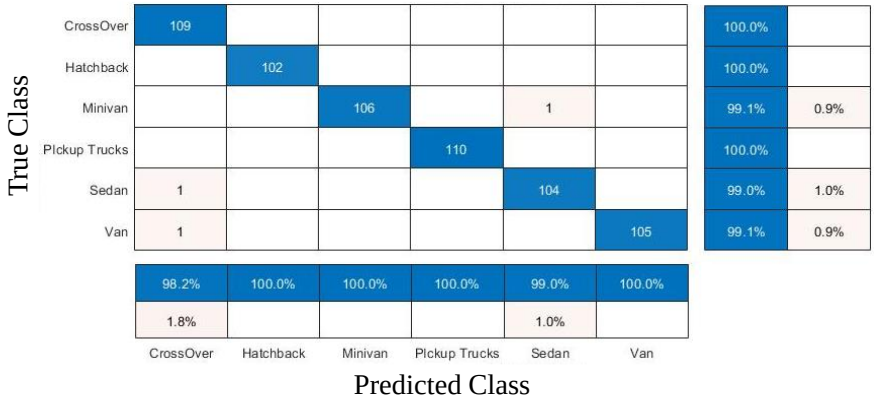

Fig. 5.1.3. The confusion matrix of the validation dataset in stage 4 with maximum classification Performance.

Table 5.1.1. shows the configuration of the training dataset used in the staged training. Stage numbers are 0, 2, 3, 4, and 5, which means how many times the training dataset instances are duplicated. Stage 0 is used for the original training dataset, stage 2 is a 2 times duplication of the original dataset, stage 3 is a 3 times duplication, and so on. We continued in this staged training until the optimum classification was obtained. Fig. 5.1.3. shows the confusion matrix of the testing dataset classification results in stage 4, where the accuracy reached the maximum value.

5.3 Experimental results

In the experiments, we evaluated the ResNet-50 based model in several consecutive stages. The experimental results are shown in Table 5.2.1.

Tab. 5.2.1. Classification performance in 5 stages of duplication.

Class / Stages	Crossover	Hatchback	Minivan	Pickup	Sedan	Van	**Average Accuracy [%]**
0	91.3	95.8	87.9	98.1	85.8	97.2	92.68
2	97.3	98.1	98.1	100	99.0	100	98.75
3	97.3	98.1	98.2	100	100	100	98.93
4	**98.2**	**100**	**100**	**100**	**100**	**100**	**99.70**
5	98.1	99.8	99.7	99.9	100	100	99.58

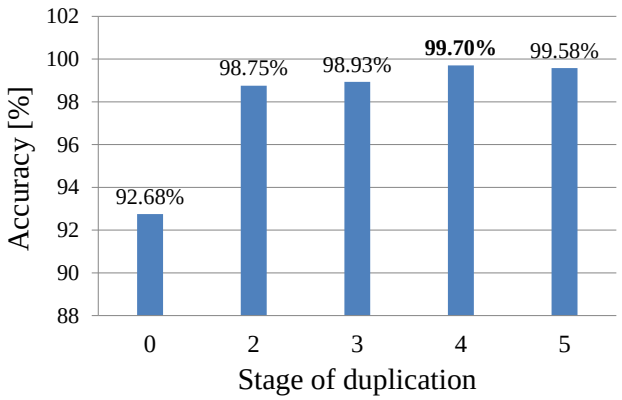

Fig. 5.2.1. Comparison of classification accuracy in 5 stages. Stage 0 is the baseline data augmentation without any duplication, 2, 3, 4, and 5 are the applied proposed data

Fig. 5.2.1. summarizes the overall accuracy for all 5 stages. Overall accuracy in stage 0 is 92.68 % without any duplication. The maximum accuracy in all stages is 99.70 % and achieved in stage 4. It is noted that the accuracy increases by increasing the number of duplicates and then starts to decrease when the number of duplicates is more than stage 4 because the model then starts overfitting. Fig. 5.2.2. shows the experimental results of vehicle classification when applying the proposed method. It shows samples of Minivan, Van, Sedan, Crossover, and Pickup vehicles that are successfully classified.

Fig. 5.2.2 Experimental results of vehicle classification with the proposed method applied in different stages.

5.4 Generalization

The ResNet-50 based proposed model was tested on Stanford based dataset consisting of 8,000 images grouped into 6 categories. The objective is to prove that the network not only behaves properly with the testing dataset but also behaves similarly with a real-world dataset. We collected, visually, 8000 vehicle images for the 6 categories of interest from the Stanford dataset and grouped them manually into those categories We retested the model, whose training dataset instances were duplicated 4 times, on this constructed Stanford- based dataset and recorded the classification accuracy of each of the six categories and the overall accuracy which is the average of the six accuracies. The average was calculated by dividing the summation of the accuracies by 6. Table 5.3.1. shows that the overall average accuracy of the model is 99.5% when tested on the Stanford-based dataset, which is very close to the lab result of 99.7%.

We used the Stanford dataset because it contains 16,185 images that are divided into 196 different categories, each category has a visually distinctive view. Every vehicle image in every category constitutes a vehicle in the foreground as well as background and different angles' views.

Tab. 5.3.1. Experimental results of accuracy applied the proposed method with ResNet-50 on Stanford 8,000 image dataset.

Class / Network	Cross-over	Hatch-back	Mini-van	Pick-up	Sedan	Van	**Average Accuracy [%]**
ResNet-50	100	99.0	98.8	99.8	99.2	100	**99.5**

Besides, they are different from each other in terms of proximity or distance from the camera, the intensity of illumination, etc. Also, some of the images are of high quality, indicating that they were taken professionally, while others are of relatively low quality, indicating that they were collected from the Internet. Thus, it represents the real world in terms of images' characteristics we get from the street-mounted cameras.

5.5 Discussion

In general, vehicle classification needs large volumes of labeled images for training the DNN model. Training the model with very few images dataset is challenging. It means that it has access to only a limited number of data which is a cause of over-fitting. The overlearning on the training data makes the model incapable of making relevant predictions on new images. In this case, the classification effectiveness would be poor when verifying the performance of the model on a testing dataset. This problem is often solved by artificially increasing the size of the training dataset through data augmentation techniques.

In the proposed method, we enhanced the classification performance by performing a novel augmentation technique, based on data duplication, before implementing the known basic data augmentation. The training dataset instances were duplicated in each stage. Therefore, the size of the dataset is artificially increased and the learning process is improved since the proposed data duplication technique obtains more image variations in the training dataset. This improves the ability of the model undergoing training to use what it has learned in predicting new images. In the experimental results, the accuracy continues to be enhanced as the number of duplicates increases until the saturation state is reached. i.e. duplication continues to result in more new image variations in the training dataset. The reason is that, in that case, each duplicated instance is transformed into many new variants different from the original ones. Then, when increasing duplicates to more than four, the produced variants have become redundant of old ones and there are no new image variants

anymore. The model then, with duplicates more than 4, exhibits over-fitting as it is thus become adapted to the redundant variants specifically. In the custom dataset, vehicle categories are balanced in terms of the number of images in each category as shown in Fig. 5.4.1., otherwise, the classification process will be biased or over-fitted towards categories of a high number of instances.

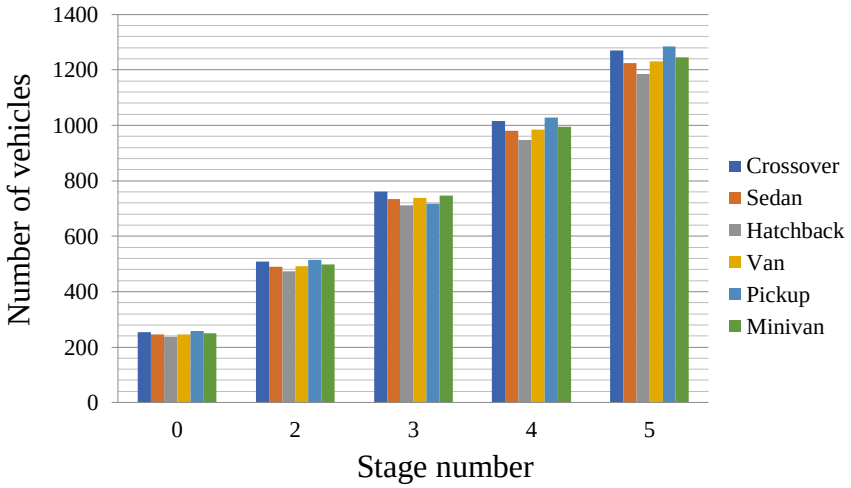

Fig. 5.4.1. Balanced sampling in the 5 stages.

This is a common problem in machine learning related to sample balancing. ResNet-50 was chosen as a baseline network because it is less demanding in terms of computation resources. Besides, according to [65], if our target is to reach the most accuracy larger than 75 %, ResNet-50 achieves the maximum throughput, (i.e. the number of inferences per second) when compared with other DNNs. To ensure the generalization of the proposed method, the model was re-tested on a Stanford-based dataset of 8000 images grouped into the six categories where we get, nearly, the same classification performance with ResNet-50 of 99.5 % as shown in Table 5.4.1. Other vehicle classes would be classified as “others” according to the evidence reversal concept, where the inexistence of any vehicle in any of the 6 classes is used as a clue of its existence as one of the unauthorized classes. The performance of the proposed method is compared with some state-of-the-art vehicle classification methods as shown in Table. 5.4.1. The comparison shows that the proposed method outperforms them. In the comparison table, accuracies of methods verified on custom datasets are considered lab accuracy. In [50], the achieved lab accuracy is 99.68 % which is very close to our results. However, when the model was tested on VeRi dataset, the accuracy was 97.66 %, with an accuracy gap of 2.02 %. In our method, lab accuracy is 99.7 % and real-world accuracy is 99.5 %. The gap is only 0.2%.

Tab. 5.4.1. Comparison of accuracy of existing methods.

Method	Dataset	Lab Accuracy	Real-world accuracy
S. Y. Jo et al. [47]	Limited scale vehicle dataset	98.30%	-
A. T. Sasongko et al. [48]	Dataset collected by a smartphone camera	99.00%	-
Z. Dong et al. [49]	Custom dataset	95.70%	-
M. Atif Butt et al. [50]	VeRi dataset	99.68%	97.66%
L. Zhuo, et al. [51]	Custom vehicle dataset including 13,700 images of six classes	98.26%	-
H. Jung, et al. [52]	MIO- TCD traffic dataset	-	97.90%
Our work	**8,000 images from Stanford dataset**	**99. 70%**	**99.50%**

Besides, in [50], the model is based on ResNet-152, which is much deeper, i.e. has a higher computational cost, and with higher inference time when compared with ResNet-50. This makes our model more proper for real-time implementations. Additionally, Network training in [71] was performed on a machine equipped with the RTX 2080TI,11 GB DDR5 GPU, core i9 – 9900k CPU with 32 GB RAM. It took 8 hours to complete the training. In our model, the training was implemented on a GPU-enabled PC with the following specification: NVIDIA GeForce RTX 2080 Ti, Processor Intel Core i7-9700 CPU@300 GHz, 3000 MHz, 8 Cores, and 8 logical processors. It took 4 hours to complete the training.

Our model outperforms [52] as well in the real-world test accuracy. However, the model in [52] is based on ResNet-18 which is more proper in real-time implementations as it is less deep and with less inference time, when compared with ResNet-50 [66]. It is worth mentioning that the comparison doesn't include [53, 54] as they used different benchmarks. They used precision and recall, rather than accuracy, to assess their method because of the imbalanced nature of the dataset. In [53], balanced sampling is used to augment, and thus alleviate the unbalanced data. They trained an ensemble CNNs based architecture on this augmented training data set and got an enhanced mean precision of all categories in case of high overall accuracy. In [54] They used a new image synthesis technique based on GANs. Through this technique, rare classes of training data are augmented with synthetic images.

5.6 Contribution

There are many contributions in this work as the following.

1. An empirical proof was provided to show that vehicle classification is enhanced in DNN by implementing a technique of augmenting the training dataset by duplicating the data instances.
2. The proposed method makes use of the duplicated data, which is usually avoided or reduced when conducting training of DNN based computer vision systems. In our method, the training data is only intentionally duplicated many times to enhance the data augmentation and thus transformed this phenomenon from being very negative to be a good technique in similar Deep Learning systems.
3. The evidence reversal phenomenon was highlighted throughout this work. According to this phenomenon, the inexistence of any object in a certain category is used as a clue of its existence in the opposite one. Specifically, six specific classes of interest are distinguished and classified. Thus, by definition, all vehicle classes, other than the 6 classes of interest, are considered as classes of “no interest” i.e. unknown. In practical implementations, where we are focusing on monitoring traffic rules in a certain area, these classes of no interest are temporarily considered violating rules until proven otherwise. That is to say, when you classify 6 specific classes, the unclassified classes are considered unknown and needed to be checked for a potential rules violation
4. A good example of using Transfer Learning, through which a DNN model focuses only on a limited number of objects, was demonstrated. In this example, a modified ResNet-50 network was trained and tested on a small custom dataset and re-tested further on 8000 images to ensure the capability of the generalization.
5. A comparison between the proposed vehicle classification method and the existing ones was conducted to highlight the effectiveness of the proposed method.

CHAPTER 6

Conclusion and Future work

In this chapter, we look back at how the thesis has emerged and how the results were reached. We presented a large picture of the thesis and summarized it. Additionally, the important findings in our work are highlighted where we showed how they have emerged from the whole work. we showed also that even though we have focused on this object of interest, which is vehicles, the presented method is easily transferable to other Deep Learning-based solutions. In the end, we highlighted the future work of this thesis. In particular, the use of augmentation by duplication in the fine-grained classification of vehicles.

In the thesis, chapter 1 is titled “Introduction” where the significance and application potential of the vehicle classification is highlighted, the thesis brief is presented and a summary of the contributions of the work is pointed at. Chapter 2 is titled “Background” where we investigate and discussed several topics and technologies which are, one way or another, relevant to our work. It deals first with how we could use computers to acquire an understanding of digital images and how to automate the tasks that are accomplished through human vision. Then, it investigates the field of Deep Learning. We also describe the concepts of Transfer learning and data augmentation which are used heavily in our work. The chapter ends up with image classification. Chapter 3, titled Related work, where we present the works related to our work. Namely vehicle type classification and vehicle fine-grained classification. It ends up with a discussion and analysis on related works to point at what research gaps that this study fills in. Chapter 4, titled “Proposed method”, where we mainly focus on how the proposed method is going to solve the research problem. We highlighted the concept of the pre-trained networks and focused on ResNet-50 specifically which we adopted in our method. Additionally, we described how we used data duplication, to enhance vehicle type classification and showed how duplication is performed only on the training dataset so that the testing dataset contains unique samples and, accordingly represents the real-world data. We also

described how the dataset was organized and how we split it into training, validation, and testing sets. The chapter ends up with describing the model training and how the proposed augmentation technique affects the training process and defines the criterion of ending the training. Chapter 5, titled “Experiments”. In this chapter, we described how staged training was performed. We highlighted the baseline training, in which the basic data augmentation was implemented, and depicted the consecutive training sessions that follow the baseline one. We also showed how the optimum classification is identified in the experimental results. Furthermore, to ensure the model generalization, we highlighted testing the model on the Stanford image dataset which represents the real-world data. Then, there is a discussion about the experimental results where we showed how they fulfilled the work goal and how the learning process was improved and the classification performance was enhanced by performing the proposed augmentation technique based on data duplication. At the end of this chapter, a comparison between the performance of the proposed method some state-of-the-art vehicle classification methods was depicted. Chapter 6, titled “Conclusion and future work”. In that chapter, we can notice how we presented the large picture of the thesis through which we can see how the important findings in our work have emerged from the whole work. In particular, we highlight over here the new approach for enhancing vehicle classification through which we enhance training data augmentation by duplicating the training dataset instances.

6.1 Conclusion

In this work, the problem of vehicle type classification was addressed. The proposed approach is based on using data duplication to artificially increase the training data set. The goal of this research is to build a vehicle type classification method, making advantage, on the one hand, of data duplication as a novel augmentation technique through which the training data set is artificially increased to a level that makes the lab performance very close to the real world one. On the other hand, Transfer Learning is used, through which we presented a Deep Neural Network model focusing only on a limited number of objects. In this work, a ResNet-50 network was trained on 1488 images and tested on 640 images of different types. From a design point of view, the offered method is very effective in the sense that it benefits from each technology mentioned above. Its characteristics can be summed through the computing power offered by Neural Networks, the ability of the Transfer Learning approach to deal with the insufficiency in data, and the use of the duplicated data. For the latter, It has shown to be very beneficial and effective in terms of enhancing

the model performance.

The work carried out consists of two fundamental steps: the first comprises duplicating the training data, intentionally, many times to enhance the data augmentation and thus transformed this phenomenon from being very negative, as being usually avoided or reduced when conducting training of DNN based computer vision systems, to be a good technique in Deep Learning systems. And the second is the pure vehicle classification step. For the latter, ResNet-50 was used in the context of Transfer Learning as a baseline pre-trained classification network. In this regard, this pre-trained model was re-trained on a small dataset consisting of hundreds of labeled vehicle images belonging to different vehicle types and thus benefited from its optimal configuration achieved during the pre-training. The model is then capable of making classification predictions on the data at hand and defining the estimated accuracy. The important findings in our work have emerged from the whole work. In particular, we proved, empirically, that this technique of augmenting training data by duplication enhances the classifier performance by a considerable value. We tested the model on a custom dataset of 640 images, grouped into 6 categories, and proved that the overall classification accuracy has improved from 92.68 % without any duplications in the training dataset, to an accuracy of 99.70 % with 4 duplicates for every instance in the training dataset.

To verify the performance of the proposed method, i.e. to assure that it will perform well on the test data that represents the real world, generalization testing was implemented where the ResNet-50 based proposed model was tested on a generalization test set. That test set is a Stanford-based- dataset consisting of 8,000 images grouped into 6 categories. The objective was to prove that the model not only behaves properly with the testing dataset but also behaves similarly with real-world data in classification problems. After development, implementation of the proposed model, and analysis of the results obtained, it was observed that the model provides very satisfactory vehicle type classification performance and that its performance exceeds the state of the art. We conclude that, in the operation of classifying vehicle types to determine correctly the class to which it belongs, the use of Transfer Learning and the proposed novel duplication-based data augmentation technique enhanced vehicle type classification considerably.

6.2 Future work

In light of what has been done, the preliminary objective has essentially been achieved. However, in addition to that achievement, future ideas are emerging. We believe that the performance of the vehicle's fine-grained classification could be significantly improved if we used the same novel

duplication-based data augmentation technique. The proposed research title then is “Fine-grained Vehicle Classification Using Data Duplication and Transfer Learning in Deep Neural Networks”. That research purpose would be to provide a fine-grained vehicle classification method that is more accurate than state-of-the-art ones. In such methods, training the model with very few images dataset is challenging. It means that it has access to only a limited number of data which is a cause of over-learning and, accordingly, over-fitting. Over-learning makes models incapable of generating predictions of new images. In this case, the classification effectiveness would be poor when verifying the performance of the model on a testing dataset. Thus, the problem encountered in this regard has been that vehicle classification needs large volumes of labeled images for training the DNN model. This problem is often solved by increasing the size of the training dataset artificially through data augmentation techniques. Hence, the main idea in the future proposed research has emerged. Its main approach would be to re-train a baseline pre-trained network on a training dataset that is augmented by two consecutive techniques to achieve better performance; one of these techniques is the novel one that we have used in this work, through which we duplicate the training dataset instances. Therefore, each duplicated instance is transformed into many new variants, different from the original ones when they are augmented further by other basic augmentation techniques that follow. In those basic techniques, the training images are randomly flipped along the vertical axis and randomly translated horizontally and vertically. Accordingly, the size of the dataset would be artificially increased and the learning process would be improved since these augmentation techniques result in more training samples for the DNN model.

Bibliography

[1] Chompookham, Thipwimon & Surinta, Olarik, Ensemble Methods with Deep Convolutional Neural Networks for Plant Leaf Recognition. ICIC Express Letters. 15. 10.24507/icicel.15.06.553, pp. 553-565, 2021.

[2] A. Harras, A. Tsuji, K. Terada, Detection of Various Vehicles Using YOLO Object Detector, The International Workshop on Frontiers of Computer Vision (IW-FCV) Ibusuki, Japan, Feb. pp. 20-22, 2020.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788, 2016.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet : A large-scale hierarchical image database," in IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255, 2009.

[5] Alan M Turing. Computing machinery and intelligence. In Parsing the Turing Test, Springer, pp. 23–65, 2009

[6] Milan Sonka; Vaclav Hlavac; Roger Boyle, Image Processing, Analysis, and Machine Vision. Thomson.ISBN0-495-08252-X, 2008

[7] AI and computer vision,2021 https://letstalkscience.ca/educational-resources/backgrounders/ai-and-computer-vision

[8] Murphy, R. R., Tadokoro, S. and Kleiner, A., Springer Handbook of Robotics, Springer, pp. 1151–1173, 2016.

[9] Arnold, S., Ohno, K., Hamada, R. and Yamazaki, K.:, An image recognition system aimed at search activities using cyber search and rescue dogs, Journal of Field Robotics 36(4), 677–695, 2019.

[10] Bengio, Yoshua. Learning Deep Architectures for AI.Foundations and Trends in Machine

Learning, 2009.

[11] Expert system, https://expertsystem.com/machine-learning-definition/, 2017.

[12] Blanco-Medina, P., Fidalgo, E., Alegre, E., Vasco-Carofilis, R. A., Jañez-Martino, F. and Villar, V. F., Detecting vulnerabilities in critical infrastructures by classifying exposed industrial control systems using deep learning, Applied Sciences 11(1), 1–14,2021.

[13] W. S.McCulloch,W. Pitts., A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4) :115–133, Dec. 1943.

[14] K. Fukushima. Neocognitron A self-organizing Neural Network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4) :193–202, Apr. 1980.

[15] C. Cortes and V. Vapnik. Support-vector networks. Machine Learning, 20(3) :273–297, Sept. 1995.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F. F. Li. ImageNet a Large-Scale Hierarchical Image Database. IEEE conference on computer vision and pattern recognition, pages 248–255, June 2009.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Advances in Neural Information Processing Systems , Curran Associates, Inc., pp. 1097–1105, 2012.

[18] J. Lettvin, H. Maturana, W. McCulloch, and W. Pitts. What the Frog's Eye Tells the Frog's Brain. Proceedings of the IRE, 47(11) :1940–1951, Nov. 1959.

[19] X. Glorot, A. Bordes, and Y. Bengio. Deep Sparse Rectifier Neural Networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pages 315–323, June 2011.

[20] H.W. Lin,M. Tegmark, and D. Rolnick. Why does deep and cheap learning work so well ? Journal of Statistical Physics, 168(6) :1223–1247, Sept. 2017.

[21] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. Neural networks : Tricks of the trade., pages 437–478, June 2012

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15 :1929–1958, 2014.

[23] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E.Howard, W. Hubbard, and L. D. Jackel. Back propagation applied to handwritten zip code recognition. Neural computation, 1989.

[24] Search Enterprise AI,

https://searchenterpriseai.techtarget.com/definition/convolutional-neural-network

[25] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. arXiv preprint arXiv :1603.07285.,Mar. 2016.

[26] An intuitive guide to Convolutional Neural Networks, https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/

[27] M. D. Zeiler, G. W. Taylor and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," International Conference on Computer Vision, doi: 10.1109/ICCV.2011.6126474, pp. 2018-2025, 2011.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11) :2278–2324, Nov. 1998.

[29] Goodfellow I., BengioY., And Courville A. Deep Learning, (Adaptive Computation and Machine Learning series), https://www.deeplearningbook.org/

[30] SinnoJialin Pan and Qiang Yang, “A Survey on Transfer Learning ”. IEEE Transactions on Knowledge and Data Engineering , (22) 10. 1345-1359, 2010.

[31] S. J. Pan and Q. Yang, A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2009.191. vol. 22, no. 10, pp. 1345-1359, Oct. 2010.

[32] W. Dai, Q. Yang, G. Xue, and Y. Yu, “Self-taught clustering,” in Proceedings of the 25th International Conference of Machine Learning. ACM, pp. 200–207, July 2008.

[33] S. J. Pan and Q. Yang, A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2009.191. vol. 22, no. 10, pp. 1345-1359, Oct. 2010.

[34] S. J. Pan and Q. Yang, A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2009.191. vol. 22, no. 10, pp. 1345-1359, Oct. 2010.

[35] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

[36] David A van Dyk & Xiao-Li Meng, The Art of Data Augmentation, Journal of Computational and Graphical Statistics, DOI: 10.1198/10618600152418584, 10:1, 1-50, 2001.

[37] Shorten, C. and Khoshgoftaar, T. M.. A survey on image data augmentation for deep learning. Journal of Big Data, 6(1):60, 2019.

[38] G. P. Zhang. Neural networks for classification : a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 30(4) :451–462, 2000.

[39] Michalski, R. S.; Stepp, R. E., Learning from observation: Conceptual clustering . In

Michalski, R. S.; Carbonell, J. G.; Mitchell, T. M. (eds.). Machine Learning: An Artificial Intelligence Approach. Palo Alto, CA: Tioga. pp. 331–363, 1983.

[40] On line, https://paddleclas.readthedocs.io/en/latest/models/models_intro_en.html

[41] Annapurani. K, Divya Ravilla , CNN based Image Classification Model , International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: Volume-8, Issue-11S, 2278-3075, September 2019.

[42] ROC graphs: Notes and practical considerations for data mining researchers. Technical Report HPL-2003-4, HP Labs, 2003.

[43] Xiao Ke, Yufeng Zhang, Fine-grained vehicle type detection and recognition based on dense attention network, Neurocomputing, Volume 399, Pages 247-257, ISSN 0925-2312, 2020. https://doi.org/10.1016/j.neucom.2020.02.101

[44] M. AbdelMaseeh, I. Badreldin, M. F. Abdelkader and M. El Saban, Car Make and Model recognition combining global and local cues, International Conference on Pattern Recognition (ICPR), pp. 910- 913, 2012.

[45] J. Fang, Y. Zhou, Y. Yu, and S. Du, Fine-Grained Vehicle Model Recognition Using A Coarse-to-Fine Convolutional Neural Network Architecture, in IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2016.2620495, vol. 18, no. 7, pp. 1782-1792, July 2017.

[46] Joya C, Li S. ZLCC: Vehicle Detection and Fine-Grained Classification Based on Deep Network Responses and Hierarchical Learning. In ITITS 18 pp. 350-360, Aug 2017.

[47] S. Y. Jo, N. Ahn, Y. Lee, and S. Kang, "Transfer Learning-based Vehicle Classification," 2018 International SoC Design Conference (ISOCC), Daegu, Korea (South), doi: 10.1109/ISOCC.2018.8649802, pp. 127-128, 2018.

[48] A. T. Sasongko and M. Ivan Fanany, Indonesia Toll Road Vehicle Classification Using Transfer Learning with Pre-trained ResNet Models, International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), doi: 10.1109/ISRITI48646.2019.9034590, pp. 373-378, 2019.

[49] Z. Dong, Y. Wu, M. Pei, and Y. Jia, Vehicle Type Classification Using a Semi-supervised Convolutional Neural Network, IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, doi: 10.1109/TITS. 2402438, pp. 2247-2256, 2015.

[50] Butt, Muhammad & Khattak, Asad & Shafique, Sarmad & Hayat, Bashir & Abid, Samia & Kim, Ki-Il & Ayub, Muhammad & Sajid, Ahthasham. Convolutional Neural Network Based Vehicle Classification in Adverse Illuminous Conditions for Intelligent Transportation

Systems. Complexity. 10.1155/2021/6644861,pp. 1-11, 2021

[51] Zhuo, L., Jiang, L., Zhu, Z. et al. Vehicle classification for large-scale traffic surveillance videos using Convolutional Neural Networks. Machine Vision and Applications 28, 793–802 2017. https://doi.org/10.1007/s00138-017-0846-2

[52] H. Jung, MK Choi, J. Jung, JH Lee, S. Kwon, WY Jung, ResNet-Based Vehicle Classification and Localization in Traffic Surveillance Systems, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 61-67, 2017.

[53] W. Liu, M. Zhang, Z. Luo, and Y. Cai, An Ensemble Deep Learning Method for Vehicle Type Classification on Visual Traffic Surveillance Sensors, IEEE Access, vol. 5, doi:10.1109/ACCESS.2017.2766203, pp. 24417-24425, 2017.

[54] W. Liu, Z. Luo, S. Li, Improving Deep Ensemble Vehicle Classification by Using Selected Adversarial Samples, Knowledge-Based Systems, Vol. 160, ISSN 0950-7051, pp. 167-175, 2018.

[55] Y. LeCun et al., "Backpropagation Applied to Handwritten Zip Code Recognition," in Neural Computation, vol. 1, no. 4, doi: 10.1162/neco.1989.1.4.541, pp. 541-551, Dec. 1989.

[56] SearchEnterprise AI, https://searchenterpriseai.techtarget.com/definition/convolutional-neural-network

[57] I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016.

[58] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11) :2278–2324, Nov. 1998.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15 :1929–1958, 2014.

[60] Tan C., Sun F., Kong T., Zhang W., Yang C., Liu C. ,A Survey on Deep Transfer Learning. In: Kůrková V., Manolopoulos Y., Hammer B., Iliadis L., Maglogiannis I. (eds) Artificial Neural Networks and Machine Learning – ICANN 2018. ICANN 2018. Lecture Notes in Computer Science, vol 11141. Springer, Cham, 2018. https://doi.org/10.1007/978-3-030-01424-7_27

[61] J. Krause, M. Stark, J. Deng, L. Fei Fei, 3D Object Representations for Fine-Grained Categorization, 4th IEEE Workshop on 3D Representation and Recognition at ICCV 2013, 3dRR-13, 2013.

[62] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

[63] Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. Journal of Big Data, 6(1):60. 2019.

[64] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, A Simple Way to

Prevent Neural Networks from Overfitting. J. Mach. Learn. Res. 15, pp. 1929-1958, 2014.

[65] Bianco, Simone & Cadene, Remi & Celona, Luigi & Napoletano, Paolo, Benchmark Analysis of Representative Deep Neural Network Architectures, 2018.SF