An Approach to One-shot Identification with Neural Networks

Janis Mohr[1], Finn Breidenbach[2] and Jörg Frochte[1]
[1]*Interdisciplinary Institute for Applied Artificial Intelligence and Data Science Ruhr, Bochum University of Applied Science, 42579 Heiligenhaus, Germany*
[2] *Trimet Aluminium SE, Aluminiumallee 1, 45356 Essen, Germany*

Keywords: Machine Learning, One-shot Identification, Image Recognition, Data Augmentation, Convolutional Neural Networks.

Abstract: In order to optimise products and comprehend product defects, the production process must be traceable. Machine learning techniques are a modern approach, which can be used to recognise a product in every production step. The goal is a tool with the capability to specifically assign changes in a process step to an individual product or batch. In general, a machine learning system based on a Convolutional Neural Network (CNN) forms a vision subsystem to recognise individual products and return their designation. In this paper an approach to identify objects, which have only been seen once, is proposed. The proposed approach is for applications in production comparable with existing solutions based on siamese networks regarding the accuracy. Furthermore, it is a lightweight architecture with some advantages regarding computation coast in the online prediction use case of some industrial applications. It is shown that together with the described workflow and data augmentation the method is capable to solve an existing industrial application.

1 INTRODUCTION

Mammals have the ability to recognise patterns and acquire new skills. In particular, humans can decide, whether two objects belong to the same category. For example, humans can directly recognise that two different balls belong to the same category, but a ball and an apple do not. This is an essential survival skill for recognising dangerous individuals and environments and for identifying food. People acquire such skills very quickly and are able to do so without ever having seen balls or apples before (Kühl et al., 2020). They use their previously acquired knowledge to make a specific guess.

Machine learning, on the other hand, is currently often trained on very large amounts of data but still may fail, when it is shown new and previously unknown data. Unlike tracking the position of a known object such as a sphere, as done in (de Jesús Rubio et al., 2021), where the technologies can use more data regarding a specific object, the application here is quite different. A particularly interesting challenge is the training with very little available data. One-shot learning means, that an image of a class is only seen once. A special case is one-shot identification. The task here is to decide, whether two objects belong to the same class or not. A neural network could, for example, be shown images of an object during a sequence of different manufacturing steps, and it would then be able to identify the object.

Computer vision methods can be used for non-invasive inspection of the manufacturing output. Therefore, objects that are very fragile or need to be processed under special circumstances, which would make it impossible to attach a tag, can be identified. This solution can be added to already existing assembly lines and manufacturing processes. It does not increase the manufacturing time and conserves energy. The use-case reported in this paper presents another case, where tags can not be used, because they would be destroyed during the manufacturing process, where the anodes are burned in a furnace. A neural network learns the deterministic variances of the surface texture to be able to distinguish objects.

1.1 Related Work

Extensive work has been done to identify objects by their structure. (Minderer et al., 2019) extracts object structures out of video clips and use them to train a neural network. Using videos would not be cost effective and difficult to implement in an indus-

Mohr, J., Breidenbach, F. and Frochte, J.
An Approach to One-shot Identification with Neural Networks.
DOI: 10.5220/0010684300003063
In *Proceedings of the 13th International Joint Conference on Computational Intelligence (IJCCI 2021)*, pages 344-351
ISBN: 978-989-758-534-0; ISSN: 2184-2825

trial application. Classic computer vision has been applied for the detection of damage and cracks in concrete surfaces. An early work, in which computer vision is used to recognise damages in concrete, contains a comparative study of various image processing techniques including fast Haar transform and Canny edge detector (Abdel-Qader et al., 2003). (Cha et al., 2017) presented an approach to detect defects on concrete surfaces based on machine learning. A deep convolutional neural network was used and compared to various techniques from classical rule-based machine vision like Canny edge detection. They pointed out, that neural networks can outperform classic machine vision techniques, which were previously used for analysing concrete surfaces. Additionally, they worked out, that Convolutional Neural Networks (CNN) are very robust and can handle various lighting situations. (Mahieu et al., 2019) already looked at tracking carbon blocks. They based their work on vision systems and tried to identify objects on assembly lines during production with cameras. Object tracking in an industrial environment already occupied (Benhimane et al., 2008), which came up with an approach to track 3D objects in real-time based on a template management algorithm. The proposed approach is much more straightforward and more accessible making it easier to implement for industrial applications. (Brusey and Mcfarlane, 2009) propose to use RFID chips to identify objects, which need to be attached to the product and could be destroyed during the manufacturing process making it impossible to automatically track an object.

Identifying objects is an important task in industrial context, and machine learning techniques are used increasingly in industrial machine vision tasks. One-shot learning and especially one-shot identification has received limited exposure and research in machine learning publications. In spite of that some work exists. Although the idea of siamese networks has been around for quite some time (Bromley et al., 1993), it has only recently been used for one-shot learning (Chicco, 2020). (Koch et al., 2015) use siamese neural networks for one-shot image recognition in which they use two identical neural networks, which are trained and then fed with two different images. The task is to determine, if those two images belong to the same class. (Deshpande et al., 2020) especially used siamese networks to detect defects in steel surfaces during manufacturing somewhat similar to the industrial application presented in this paper. Siamese networks can also be used for robust face recognition and verification as (Chopra et al., 2005; Taigman et al., 2014) have shown.

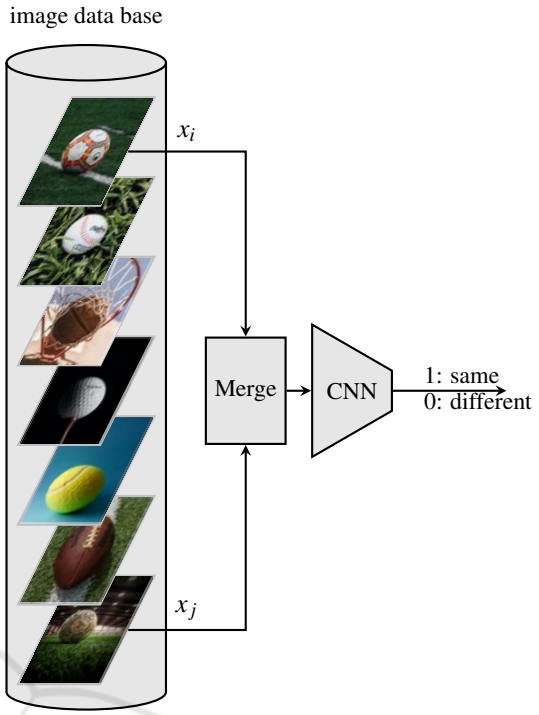

Figure 1: Proposed general approach to one-shot identification. Two images are merged and then put into a CNN to classify whether they contain the same objects or different objects.

Our contributions in this paper are summarised as follows: (1) We propose an approach for identifying objects, which have only been seen once. (2) We focus on low cost, resource-efficiency and a straightforward implementation of machine learning techniques. (3) We propose a workflow for an industrial application and show, that we achieve a high accuracy.

2 APPROACH

Although deep learning has shown great promises for image recognition tasks, it comes with the cost of large data requirements. Since it is not always possible to gather large amounts of data due to e.g. information privacy, the possible applications for machine learning are still limited. To address this issue, there has been recent interest in the research community to develop neural networks, that can effectively learn from a small amount of data. One-shot learning is one idea to reduce the amount of needed training data while still maintaining robust and accurate predictions. The key idea behind one-shot image recognition is that given a single sample of the image of a particular class, the neural network should be able

to recognise, if the candidate examples belong to the same class or not. The network learns to identify the differences in features of the input image pair in training. The simple network architecture used in this work is shown in figure 8. The proposed approach to one-shot identification can be used with every convolutional neural network, which suits the use-case. The model, once trained, should be able to identify objects despite changes in hue or surface texture. The output of the network is [same, different]. *Same* means that both images show the same object, while *different* means that they show different objects. The proposed approach aims to act in the special case, that only a very small amount of data is available for training, and images are only seen once for comparison.

Figure 1 shows the proposed approach with an exemplary data set. A convolutional neural network is trained on a data set that consists out of images of game balls used in different sports and games. The task is to compare two images of soccer balls, which have never been seen before, and to decide if they show the same type of ball or not. The network is trained with merged and labelled images, which show the same or different balls. The meaning of *merged* in practice in this method is discussed in section 3.1.1. Therefore, the neural network learns to discriminate between different types of balls and can generalise this information to compare images of objects it has never seen before. Every single image of a ball is handled as a new case in the proposed approach.

2.1 Industrial Application

The exemplary industrial application used is the automatic production of anodes in an aluminium plant. The production of these anodes is costly and time-consuming. The anodes are burned in a kiln in one of the production steps. During this process, the anodes lie irregularly on a conveyor belt. For quality assurance and traceability of defects, it is important, that the anodes can be identified between the initial jogging and the final electrolysis.

The surface structure of an anode does change while being baked. Additionally it is possible, that areas are damaged during this process through burn-off. Also the material, that is used to stabilise the anodes during a baking process, can stick to the anode surface. In both cases only certain parts of the anode are affected. Features and surface texture in other areas remain unaltered. In case of tags being used it could happen, that one of the aforementioned processes destroys the attached tag, which render them useless in terms of object identification and make this specific anode unrecognisable.

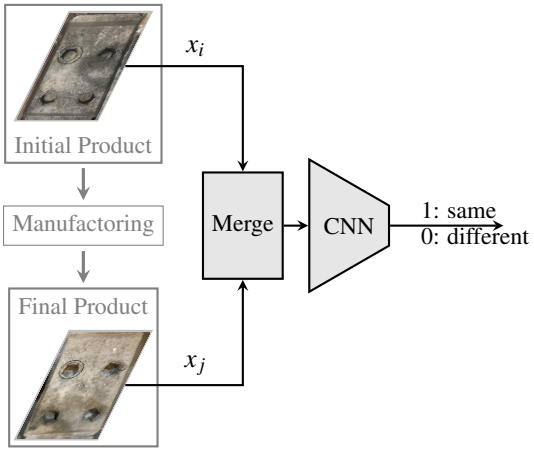

Figure 2: Diagram showing the proposed workflow for the exemplary industrial application. Images of the initial and the finished products are taken. Between the initial and finished product several manufacturing steps are done, which may alter features like appearance, surface texture and hue. The two images are merged – see section 3.1.1 for details – and fed into a pretrained convolutional neural network.

The anodes move through the manufacturing process once and therefore not enough data is generated to train a neural network to identify each anode (every anode is a class for itself). This results in the problem that a big number of anodes would need to be learned with only a single image of every anode available for training. Because of this, a neural network is to be trained to compare two anodes and make an educated guess, if both anodes are the same or not. This allows to create a neural network, that does not need to be retrained for every anode, thus solving the problem of having not enough training data. Only two images of every anode are required to identify an object.

There are pictures of every anode made on the conveyor. Whenever an anode passes through the furnace a new image is made and fed into the neural network. The neural network compares this with every other anode and outputs how confident it is, that both anodes shown to the network are the same. As the network only compares two images, it needs to be passed through several times, until the processed anode has been compared to all anodes, that have already been seen. After all anodes have been seen, it is evaluated, which two anodes have the highest likelihood. Only a few anodes pass the cameras at a time, which is very different to applications regarding face recognition. Figure 2 explains the whole process flow.

Figure 3: Variances in surface texture before (left) and after (right) the anode is baked in the furnace. Notice that the anode is lighter afterwards, which is the case at every baking.

2.2 Image Acquisition and Data Augmentation

Since the anode images for training the system are not yet available, they must first be acquired. For this purpose, first the hardware and software used and afterwards pre-processing of the images is explained.

A set-up using a Raspberry Pi 4B is used to capture the anode images. This generates an image of an anode after a movement has been detected. Since the anodes always pass the camera individually on the conveyor belt, it is ensured, that only one anode is visible in an image. A camera with an IMX477R CMOS sensor is used. It has a resolution of 12.3 megapixels. In addition, a 16mm telephoto lens is used.

First, the camera image is read-in using the open-source library OpenCV (Mahamkali and Ayyasamy, 2015), and then the influences of the environment can be observed and eliminated. The pictures taken are converted to greyscale images, because the additional colour channels do not carry any additional information in this use-case. Various machine vision techniques like edge detection are used to locate the anode in the image. Afterwards the image is reduced to the section, that shows the anode, and all images are resized to 530×330 pixels.

Due to long lasting processes for individual anodes, the changes of the surface textures are simulated. As a basis for the generated images a selection of various anodes before and after baking is used. Figure 3 shows a sample of such images. These images were taken during ongoing production.

As can be seen in Figure 3 significant features on the surface of the anode remain after processing. Reviewing the available pictures of anodes before and after the baking process allowed to analyse and identify the most common changes and similarities on an anode during production.

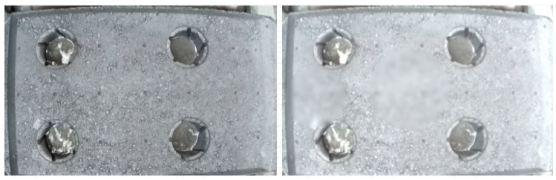

Figure 4: Generated image as it is used in the training data set. This image was generated out of a photo of an anode with data augmentation techniques.

The stubs are created during the manufacturing process of the green anode. The alignment of these always changes between two anodes, because the tool, that is in the mould to create the stub, is removed after the anode forming process. This feature remains between all processing actions. The texture and the brightness of the surface change during the baking process in the furnace. During the processing also spots and marks are added and mutilated to the anodes.

To overcome the problem of limited quantity and limited diversity of data the existing images are manipulated with affine transformations. For simulating the variances all images of unaltered anodes are manipulated. Each image in the data set is rotated randomly about its centre. Brightness and fine contours are changed. Count and position of burn-offs and build-ups are manipulated randomly and simulated with blurred circles, that are layered over the anodes. These images are saved separately and used as part of the training data set. The data set is combined out of simulated and original images. Figure 4 exemplarily shows a simulated image of an anode next to the original image.

3 EXPERIMENTAL RESULTS

A simple convolutional neural network is trained and used for the one-shot identification task. The hyperparameter values used for training the network are provided in table 1. The input to the model is a grayscale image of 330×530 pixels. Four convolutional layers with 32, 32, 64 and 64 filters are used, followed by a max pooling (Scherer et al., 2010). A kernel size of 3×3 is used for convolutions with a stride of 1. The ReLU activation function is used on the output feature maps of each layer. The convolutional layers are followed by two fully connected layers of size 128 and 2 respectively.

Table 1: Hyperparameter values used for training the convolutional neural network.

Parameter	Value
Batch size	32
Number of epochs	20
Learning Rate	1e-4
Optimiser	RMSprop (Graves, 2013)

3.1 Handover of Two Images

A conventional CNN takes one image at a time as input (LeCun et al., 1998). For comparing images, a method needs to be found, that can handle two inputs to compare two images. This leads to two different approaches. Two pictures are fused and the merged image is fed into the CNN for training. The other approach is based on using two sequences of convolution and pooling. Two pictures are given to the convolutional neural network as input. Both sequences, that are fed with different images, are concatenated in a specific layer after flattening.

The training data set, that is used to train the convolutional neural networks for the experimental results, contains original and simulated images of anodes. These are merged together randomly leading to a ratio of 50 to 70%, where the same anode is on both pictures. Furthermore, generalisation is achieved through the random sequence of images. Both merged images of real and simulated images and real images exclusively are possible.

3.1.1 Merging Two Images in Pre-processing

Several ways of merging the images are possible. One option is, that the images could be joined horizontally or vertically, as shown in figure 5. If the original dimensions are n and m the result is an $2n \times m$ or $n \times 2m$ image.

Figure 5: Merging two images into a bigger image by joining them horizontally or vertically.

The other way is to stack the images' channels. So, if one uses all colour channels, this leads to six channels, but it turns out, that for this application it is enough to use greyscale versions of the images and stack these. The process is illustrated in figure 6. Merging the images up like this leads to an architecture of the CNN as shown in Figure 8 in the appendix.

Both approaches lead to an altered tensor, that is delivered to the neural networks. Therefore, attention must be paid to the distinctions between input layer and images.

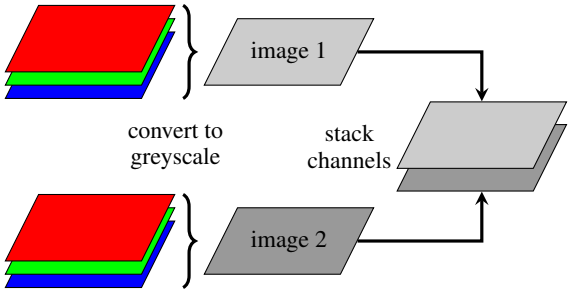

Figure 6: Merging two images converted to greyscale by stacking them resulting in a two-channel image.

At first, the influence of these two approaches is evaluated. With the same database and the same convolutional neural network the stacking approach of merging leads to a prediction accuracy of 98.36%. When joining the images, as in figure 5, they only achieve an accuracy of 61.48%.

Both were evaluated on the test data. Therefore, better results can be expected from a database with stacked images. This kind of data set will be used for training the neural network. The significant features are on the same relative position, when the colour-channels of the images are stacked. Even though a CNN can achieve a slight invariance to translations through parameter sharing, this can affect the performance of the CNN. In addition to this, the data are mixed up with the merged images because of convolutions and pooling. This does not happen with stacked images, because each action is applied to every channel separately. Therefore, the information of the original and the simulated images are not mixed up.

3.1.2 Learning Behaviour and Accuracy

Our experiments have shown, that stacking images achieves a high accuracy. Further experiments have displayed, that the convolutional neural network reaches an accuracy of 98.36%, when the early stopping terminated the training process. As 7 illustrates this high level is reached in a very stable way even showing higher results on the way up to 100%. No signs of overfitting can be found as new images are also identified correctly, and the accuracy on the validation data does oscillate strongly or even shrink during the training. Figure 7 shows the learning behaviour based on the accuracy and loss during training.

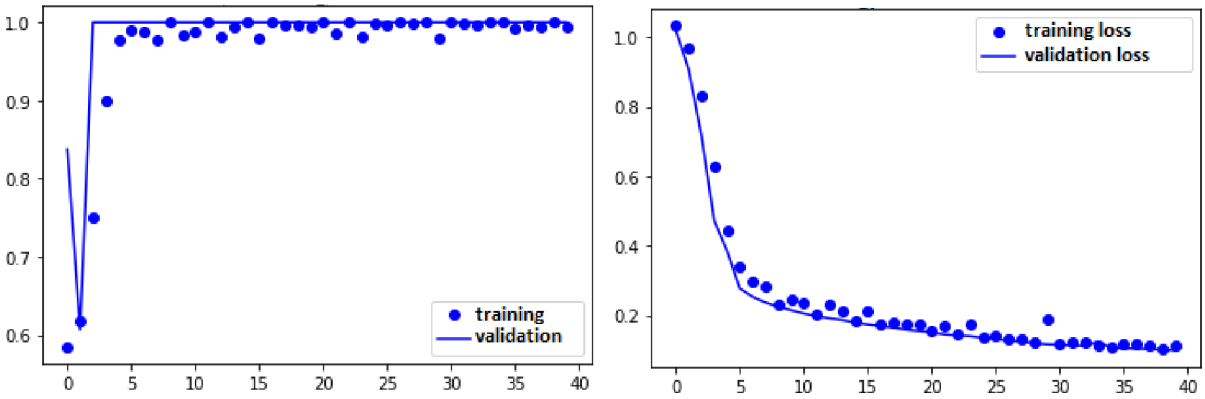

Figure 7: Development of the accuracy (left) and the loss function (right) during the training process for a neural network with one input for merged images. The changes regarding the training and the validation set are shown. The mutual and nearly monotonic behaviour underlines, that the chosen model shows good qualities regarding generalisation during the training process.

3.2 Testing the CNN

The optimised neural network is tested on its proper functionality. Therefore, pictures of anodes are used, that have not been part of the training or test data set. One original image is turned into a simulated image with the process described in section 2.2. The neural network is now fed with all original pictures in the data set (consisting out of the pictures used for training and testing and the new pictures) and the newly simulated picture. The task is to compare the new image against the data set and to find the matching anodes. On early tests with a small data set the neural network repeatedly fails to categorise images of anodes, which have burn-offs and markings on the holes. This seems to indicate, that the holes are a feature that the CNN uses to identify and compare the anodes. Mostly anodes with impureness and markings not on the holes were categorised correctly. This problem vanished after increasing the size of the data set, and the convolutional neural network was able to identify the anodes correctly, regardless of the position of burn-offs and other traces of the manufacturing process.

3.3 Siamese Networks

A siamese network as proposed by (Deshpande et al., 2020) was used to compare the results of the proposed approach on the industrial application. A siamese neural network is an architecture, in which two identical neural networks are used. They have separate inputs but share their parameters and weights. The networks have one combined output, which outputs the euclidean distance. The siamese network was trained on the same training data and the accuracy was achieved on the same set of test images as used for training the neural network in the proposed approach. It reaches an accuracy of 96%.

The proposed approach for one-shot identification has a main advantage over state-of-the-art siamese networks. Siamese networks have a more complex architecture regarding the prediction, because the images are forwarded through two neural networks instead of just one. In industrial applications with a steam of a small batch of objects to identify just once – in opposite to face recognition where siamese networks are very common – pre-processing images through the network and storing outputs and then in an additional step calculating the similarity is less helpful. Therefore, in such industrial applications they are often slower regarding prediction. Fast predictions are of the utmost importance in industrial applications, because in general they have some kind of real time requirements.

4 CONCLUSION AND FUTURE PROSPECTS

In summary we hold, that we proposed a general approach for identifying objects with machine learning techniques in a one-shot learning setting. We refined this approach to a workflow for an exemplary industrial application, in which only sparse data is available, and show that we achieve state-of-the-art accuracy. The results show, that the proposed architecture for a CNN does work properly on the exemplary use-case with anodes and might therefore be usable for similar use-cases with different objects and less significant features.

The convolutional neural networks as proposed and tested with simulated images of baked anodes can be used in a real industrial environment. As can be seen with our experiments regarding formatting and merging, the data can have a significant influence on the accuracy of a neural network.

The capturing of training data is the biggest challenge in using machine learning techniques to tackle a task of object tracking in an industrial context. To generate examples for training it is necessary to track the objects manually during production. But this method allows to track objects that can not be tracked with tags or chips because of the nature of the object or the processing steps. The collected data can also be used for tasks like predictive maintenance or quality control.

Conclusive it can be said, that the proposed neural network is the core of a system, that can track objects through identifying them in images during a manufacturing process on a production line. Future work includes the development of a data storage and acquisition solution. Furthermore, more tests should be done with different objects, and techniques of continuous learning can be tested to enhance the neural network with the capability to be retrained while already being deployed.

ACKNOWLEDGEMENTS

The work of the academic authors were funded by the federal state of North Rhine-Westphalia and the European Regional Development Fund FKZ: ERFE-040021.

REFERENCES

Abdel-Qader, I., Abudayyeh, O., and Kelly, M. (2003). Analysis of edge-detection techniques for crack identification in bridges. *Journal of Computing in Civil Engineering - J COMPUT CIVIL ENG*, 17.

Benhimane, S., Najafi, H., Grundmann, M., Malis, E., Genc, Y., and Navab, N. (2008). Real-time object detection and tracking for industrial applications. *VISAPP 2008: Third International Conference on Compter Vision Theory and Applications*, 2.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, page 737–744.

Brusey, J. and Mcfarlane, D. C. (2009). Effective rfid-based object tracking for manufacturing. *International Journal of Computer Integrated Manufacturing*, pages 638–647.

Cha, Y.-J., Choi, W., and Buyukozturk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32:361–378.

Chicco, D. (2020). Siamese Neural Networks: An Overview. *Methods in molecular biology*.

Chopra, S., Hadsell, R., and Lecun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. volume 1, pages 539– 546 vol. 1.

de Jesús Rubio, J., Lughofer, E., Pieper, J., Cruz, P., Martinez, D. I., Ochoa, G., Islas, M. A., and Garcia, E. (2021). Adapting h-infinity controller for the desired reference tracking of the sphere position in the maglev process. *Information Sciences*, 569:669–686.

Deshpande, A. M., Minai, A. A., and Kumar, M. (2020). One-shot recognition of manufacturing defects in steel surfaces. *Procedia Manufacturing*, 48:1064–1071. 48th SME North American Manufacturing Research Conference, NAMRC 48.

Graves, A. (2013). Generating sequences with recurrent neural networks.

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. *32nd International Conference on Machine Learning*.

Kühl, N., Goutier, M., Baier, L., Wolff, C., and Martin, D. (2020). Human vs. supervised machine learning: Who learns patterns faster?

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*.

Mahamkali, N. and Ayyasamy, V. (2015). Opencv for computer vision applications. *National Conference on Big Data and Cloud Computing*.

Mahieu, P., Genin, X., Bouche, C., and Brismalein, D. (2019). *Carbon Block Tracking Package Based on Vision Technology*, pages 1221–1228.

Minderer, M., Sun, C., Villegas, R., Cole, F., Murphy, K., and Lee, H. (2019). Unsupervised learning of object structure and dynamics from videos. *33rd Conference on Neural Information Processing Systems*.

Scherer, D., Müller, A., and Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. *20th International Conference on Artificial Neural Networks (ICANN)*, pages 92–101.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification.

APPENDIX

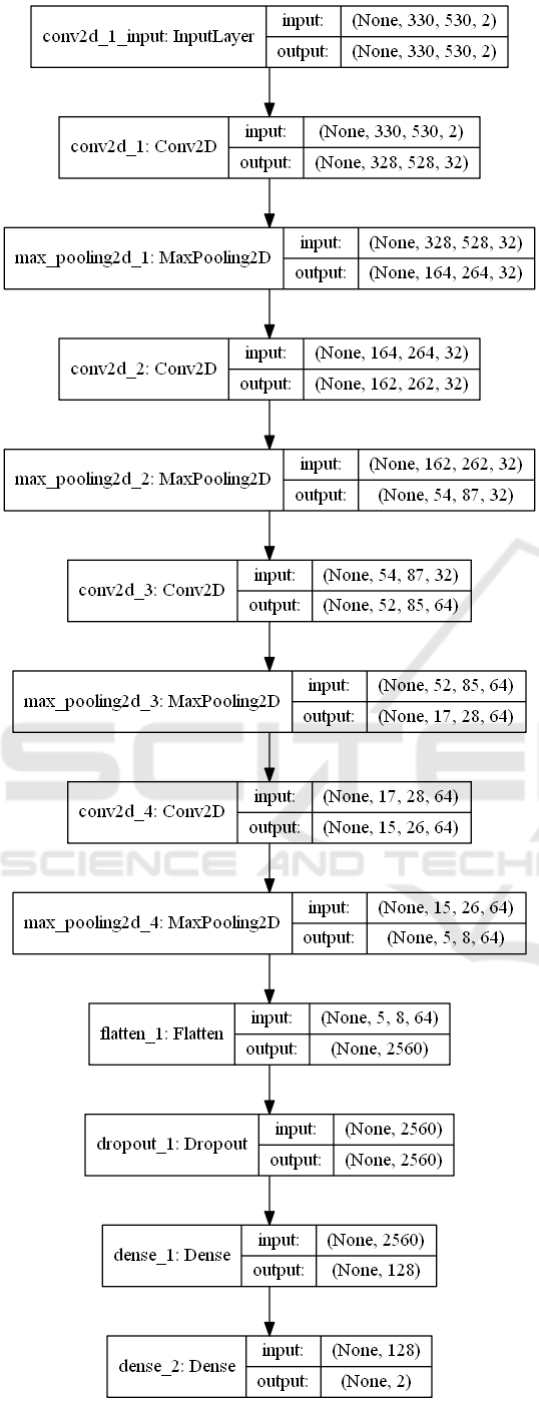

Figure 8: A paradigmatic structure of a neural network used for manually merging the pictures.