Image Classification with Convolutional Neural Networks

Yıldız Technical University/Faculty of Arts and Sciences

Department of Physics

Prof.Dr. Orhan İçelli - Hüseyin Can Kankotan

May, 2022

ABSTRACT

Using Convolutional Neural Network algorithms, which are a subcategory of Deep Learning, we attempted to process any given image dataset and classify this dataset with a certain accuracy percentage.

In this study, AlexNet, VGGnet, and ResNet algorithm architectures were implemented.

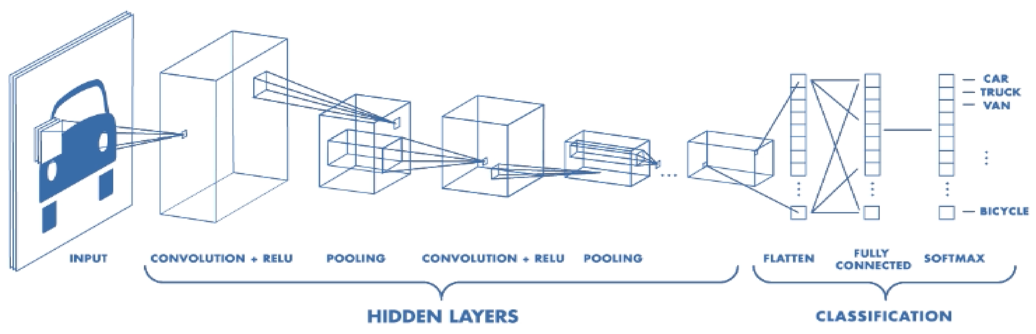
For the image dataset, the SAR (Synthetic-Aperture Radar) dataset used within the scope of the MSTAR (Moving and Stationary Target Acquisition and Recognition) program was used as a basis.

CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Network (CNN) is a machine learning model developed and continually improved by mimicking the information processing of the human brain and the synaptic connection between neurons. In human biology, neurons connect to each other in different ways to form networks. These networks have the capacity for learning, memorizing, and understanding the relationships between data. The main reason why a human can produce solutions to problems that require thinking and observation skills is the ability to learn by living or experiencing. At this point, Artificial Neural Networks approach problems like a human brain, providing solutions to the problems that humans face as a result of having these abilities.

Convolutional Neural Networks are a Deep Learning algorithm used to distinguish objects from each other in an image taken as input, or to make the desired classification based on the relationship/similarity of different images by comparing them.

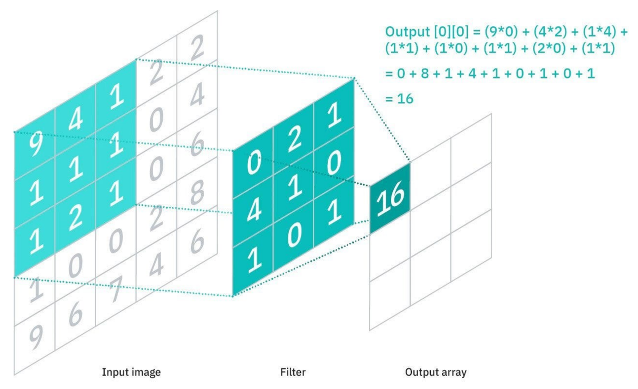
CNN can consist of different layers, but basically has 3 layers: Input, Hidden Layer, and Output. The input layer is the layer where it receives the input, i.e., an image in the dataset. The output layer is the layer where the image's class assignment is made.

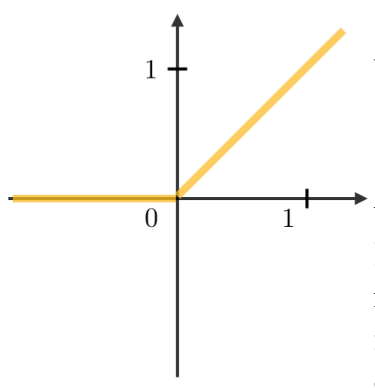


Hidden layers can vary depending on the size of the data to be processed, the desired output from the data, and the hardware on which the algorithm processing the data runs. The components in this layer are generally as follows:

- Convolutional Layer: This is the layer where the convolutional filter is applied to extract the feature map of the input image.

In the image on the right, an output matrix array is obtained by applying a 3x3 matrix filter to the input. All values are multiplied element by element in the matrix, and the sum of all values is recorded as the relevant element of the output matrix.

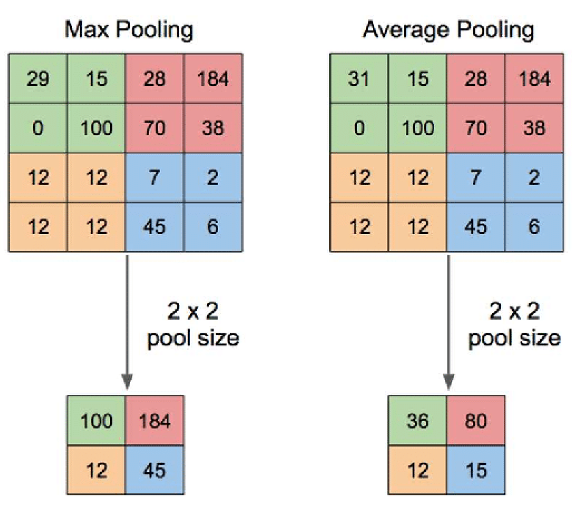




The feature map holds a feature of the input. For example, like a car's headlight. Therefore, many feature maps are needed to detect all features in the input image.

After the convolutional filter, an activation function called ReLU (Rectified Linear Unit) is applied to the feature map. This is because, since all layers can be a linear function, the neural network acts like a single perceptron; that is, the result can be calculated as a linear combination of the outputs.

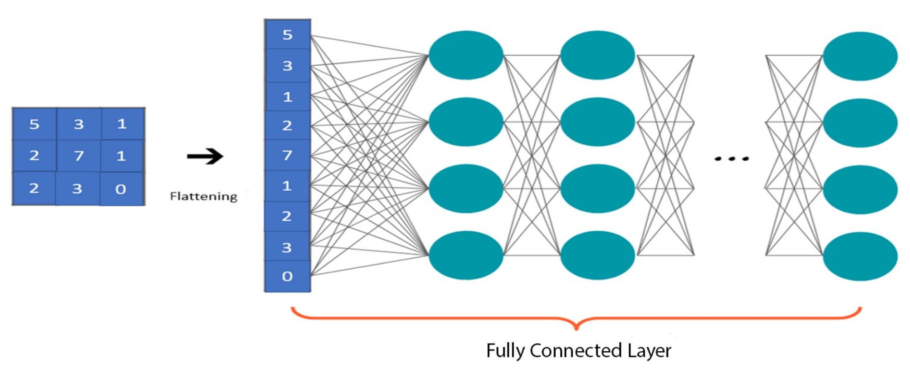
After the ReLU function, the result is 0 for values less than 0 in the output matrix, while it remains the same for values equal to or greater than 0. In the image below, black values in the feature map are negative. After applying the ReLU function to the feature map, a result like the one below is produced.



- Pooling Layer: This is a step used to reduce computational complexity. This way, both the required processing power is reduced and unnecessary features captured are ignored, focusing on more important ones.

There are generally two different pooling techniques used in CNN models. One of them is Maximum Pooling, and the other is Average Pooling.

In the pooling layer, which has a filter like in the convolutional layer, this filter again moves over the image. But this time, instead of the convolutional operation, it applies the determined pooling technique. If maximum pooling is applied, it takes the largest value in the area covered by the filter; if average pooling is applied, it takes the average of the values in the filter. As a result, the size decreases and we retain important features.



- Flattening Layer: The task of this layer is to prepare the data that will go to the last layer. The operation is to convert the matrices coming from the previous layers into a one-dimensional matrix.

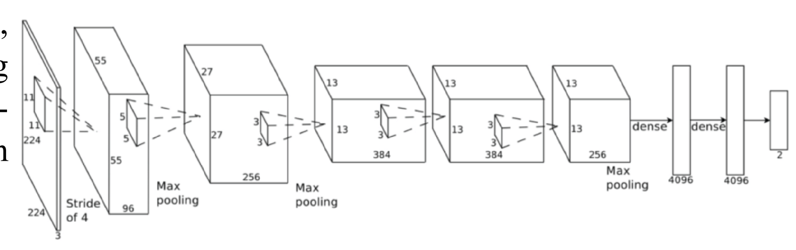
- Fully Connected Layer: This layer is the last and most important layer of CNN. Data is taken from the flattening layer and the learning process is carried out through the artificial neural network by changing the weight and bias.

CNN-BASED ALGORITHMS USED IN THE APPLICATION

1. AlexNet

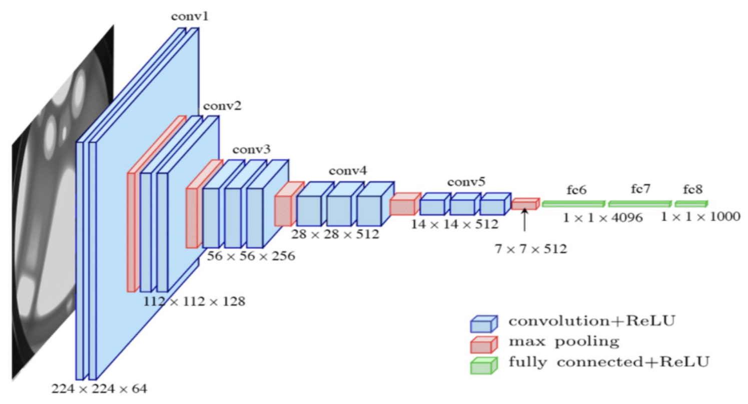
AlexNet consists of 5 convolutional layers and 3 fully connected layers. AlexNet uses ReLU as activation in non-linear parts.

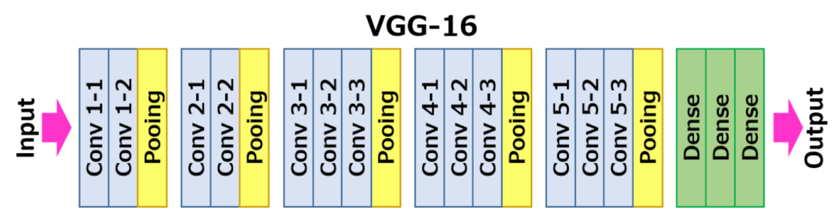
One of the problems of Deep Learning Algorithms, overfitting, that is, the algorithm memorizing the given training data instead of generalizing it, was prevented by using the dropout function with the AlexNet architecture. Thus, it was observed that thinning the nodes below a certain threshold value in fully connected layers increased performance.



2. VGG-16

VGG-16 is a CNN architecture presented by the VGG Group (Oxford). As the name suggests, it consists of 16 layers. The layers consist of 13 convolutional and 3 fully connected layers.



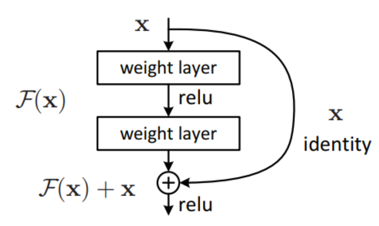


In VGG-16, the artificial neural networks architecture is simplified, reducing the high filter sizes used in the AlexNet architecture. As in other models, while the height and width dimensions of the matrices decrease from input to output, the depth value (number of channels) increases. At the output of each convolution layer of the model, filters with different weights are calculated, and as the number of layers increases, the features formed in the filters represent the depths of the image.

3. ResNet

One of the problems encountered when applying Deep Learning algorithms is vanishing and exploding gradients. When training the Artificial Neural Network, sometimes the derivatives become very small or very large, which makes it difficult for the algorithm to train data. When the gradients of the loss function with respect to the parameters in the early layers of a network are very small, it learns slowly and many gradients do not contribute much to learning because they are too small, which can lead to poor performance.

ResNet uses a technique called skip connections. With this method, it skips several layers and connects directly to the output. In this way, the vanishing/exploding gradient problem is prevented.

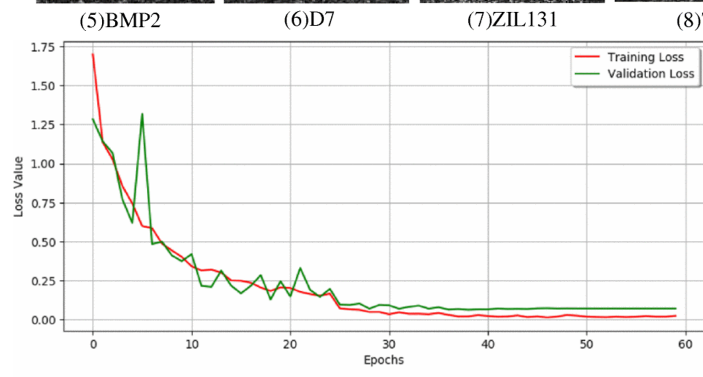


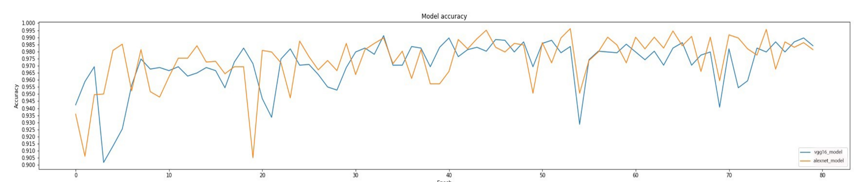
APPLICATION OF ALGORITHMS FOR MSTAR DATASET AND COMPARISON OF ALGORITHMS' EFFICIENCY

In images taken with SAR (Synthetic Aperture Radar), there is a dataset consisting of 9 different military vehicles within the scope of the MSTAR (Moving and Stationary Target Acquisition and Recognition) program. Using the Convolutional Neural Networks-based algorithms described above, this dataset was processed for each algorithm, and the classification accuracies of the algorithms for the relevant vehicles were calculated and indicated on the graph.

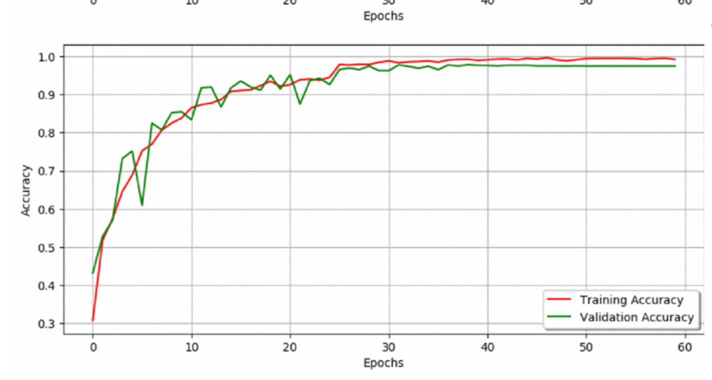
Calculations were made in Python programming language, using Keras and TensorFlow libraries.

The vehicles in the dataset and their radar images are as follows:





The vehicles in the relevant dataset were classified with an accuracy of over 95% with the applied algorithms. All 3 algorithms gave successful results.



CONCLUSION

The application areas of Convolutional Neural Networks range from detecting whether cancerous masses are benign or malignant to quickly and highly processing satellite images taken in active war zones and converting them into intelligence information.

In the developing, transforming, and evolving world towards digital, while the place of big data is important, processing this data is also quite complex, at this point Convolutional Neural Networks provide a great convenience.

REFERENCES

https://en.wikipedia.org/wiki/Convolutional_neural_network#Receptive_field, https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks, https://www.run.ai/guides/deep-learning-for-computer-vision/deep-convolutional-neural-networks#What-are-the-Types, https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, https://www.ibm.com/cloud/learn/convolutional-neural-networks, https://www.researchgate.net/publication/323587302_A_Deep_Learning_SAR_Target_Classification_Experiment_on_MSTAR_Dataset, https://arxiv.org/abs/1708.07920, https://ieeexplore.ieee.org/document/8835628, https://www.researchgate.net/publication/327878623_Military_Object_Real-Time_Detection_Technology_Combined_with_Visual_Salience_and_Psychology, https://www.researchgate.net/publication/344882909_Deep_learning_for_automatic_target_recognition_with_real_and_synthetic_infrared_maritime_imagery, https://www.researchgate.net/publication/350830115_Multiview_Deep_Feature_Learning_Network_for_SAR_Automatic_Target_Recognition, https://arxiv.org/pdf/2106.08005.pdf, https://ieeexplore.ieee.org/document/9652365, https://ieeexplore.ieee.org/document/8907833, https://www.sdms.afrl.af.mil/index.php?collection=mstar&page=targets, https://www.nsa.gov/portals/75/documents/resources/everyone/digital-media-center/publications/the-next-wave/TNW_22-2.pdf?ver=2019-02-25-104515-237, https://github.com/hamza-latif/MSTAR_tensorflow, https://github.com/rohitgit1/Target-Detection-in-MSTAR-Images, https://paperswithcode.com/paper/deep-learning-for-target-classification-from, https://apps.dtic.mil/sti/pdfs/ADA246302.pdf, https://www.sciencedirect.com/science/article/pii/0895717795002200