Object Detection using Convolutional Neural Network Transfer Learning

Seetam Emmanuel [1], F. E. Onuodu [2]

[1] Computer Science Department, KenSaro-Wiwa Polytechnic, Bori, Nigeria
[2] Computer Science Department, University of Port Harcourt, Choba, Nigeria

Abstract
Any machine learning algorithm's ability to extract salient (relevant) characteristics is critical to its success. Traditional machine learning methods rely on domain expert-generated input features or computational feature extraction techniques. A Convolutional Neural Network (CNN) is a type of artificial intelligence inspired by how the human brain's visual cortex functions when it comes to object detection. Because CNN requires a large number of neurons and layers to train data, it is not ideal for small datasets. Obtaining and storing a huge data collection for a scratch program is a challenge. These issues can be solved by using transfer training using a pre-trained data set. This is a dimensionality reduction approach used in deep learning analysis to lower the number of hidden layers and construct neural network applications on tiny data sets with high gain and little information loss. Using transfer learning to retrain a convolutional neural network to categorize a fresh batch of photos, this research investigates visual properties and isolates those that unify the digital image. The developed model satisfied 97% MSE (Mean Squared Error).

Keywords: CNN, Transfer Leaning, Pre-trained Image, Computer Vision

Introduction
The most important human sense is, without a doubt, vision. In practically every action we conduct, it is human dependability. Image recognition by computer utilizing human perception has been a bigger and difficult job in computer science, because it is quite complicated to explicate to a machine the properties that identify a certain item, as well as the way to distinguish them. The neural network can now learn these traits on its own thanks to advances in deep learning.

Deep learning applications often entail studying a domain to extract the domain pattern, which is then used to predict or categorize a new collection of domain elements. This work's fundamental anatomy is a repurposed piece of technology that found its way into CNN via an aesthetic style transfer in 2015 (I. Vasilev et al., 2019).

The use of one picture's style (or texture) to mimic the semantic content of another image is known as Artistic Style Transfer (AST). Different algorithms can be used to achieve this. In the paper "A Neural Algorithm of Artistic Style", Leon A. et al. (2015) introduced AST. It's also known as CNN-based neural style transfer (I. Vasilev et al., 2019).

Small models trained on toy datasets are straightforward and quick, but dealing with huge datasets like GoogLeNet, CIFAR-10, ImageNet, and others will take a long time and a larger network to train. Large datasets are difficult to get, and they are not always available for the tasks of interest. Furthermore, the obtained pictures must be tagged, which is both time-consuming and costly. As a result, a modest software engineer will resolve to transfer learning to tackle a genuine ML challenge with minimal resources.

Transfer learning is the process of applying a previously learned machine learning model to a new issue. A neural network built on ImageNet, for example, may be retrained to categorize goods in a grocery shop. In a similar vein, a driving simulator game may be used to train a neural network to drive a virtual automobile before applying the network to a real-world vehicle. Transfer learning is a general machine learning concept that may be applied to any machine learning technique.

Transfer training begins with a network that has already been pre-trained. In the most common case, a pre-trained net with a similar data domain is used. Popular ImageNet pre-trained neural networks may be utilized for transfer training in TensorFlow, Keras, and PyTorch. Though we have the option of training our network with any dataset we choose, but we tense to adopt reused technology.

In deep learning applications, transfer learning is frequently used to fine-tune the network to be quicker and simpler than a network that starts its training from fresh with randomized initialization weights. This method uses a pre-trained network as a starting point for learning a new task. This method used a pre-trained network as a starting point for learning the new task's features. With promising results, this strategy enhanced neural network training on a tiny data set. It uses a lower number of training objects to transfer the feature of an old trained data set to a new data set. Transfer learning is the process of retraining a convolutional neural network to categorize a new batch of pictures.

The retention procedure may be done in two ways: utilizing the original component of the network as a feature extractor and solely training the additional layer(s), or using the Fine-tuning technique. This method involves training the whole network (including the old and new layers). The advantage of this technique is that the first layers identify general features that are not related to a specific task, allowing them to be reused. The deeper layers, on the other hand, may detect task-specific traits that need to be updated. As a result, this approach was used in this study to prevent overfitting concerns.

Images databases that have been trained and preserved for research purposes are known as pre-trained images. They honed their skills on millions of photos and objects of various types and categories which the network has trained to learnt wealthy characteristics representation, with the ability to categorize an image into one of around a thousand object categories, which include items such as computers, household goods, animals, and so on.

The end layer of a CNN's fully linked layers acts as a translator between the network language (the abstract attributes pattern learnt during training) and our language, which represents each sample's class. The network takes an image as input and produces a name for each object in the image, as well as probabilities for each of the object categories.

CNN extracts features from digital objects, compresses them, and converts them to a matrix. As a result, CNNs is a feature extraction engine that has risen to prominence as a result of remarkable advances in machine learning and computer vision applications.

Many pre-trained networks, such as CIFAR-10, MIN, and others, provide the same help in many sectors of data analysis. However, based on our system design, we must chose the GoogLeNet network and the Deep Learning Model for GoogleNet as the research's basic engines.

Any physical entity that can be seen and touched is referred to as an object. Objects are distinguished by their characteristic, often known as a feature. These characteristics distinguish one thing from the others. An iconic model is a model that appears in a photograph. Object models may be created in a variety of ways, including using a camera, a sensor, or a scanner.

Digital Images are digitally recorded items, usually graphic data, that are handled by a computer system. It's the data that goes into computer vision processing. Computer vision is the idea underlying an artificial system that pulls information from pictures. In image data processing, visualization is a crucial tool. Image graphically communicates concepts to others for simple comprehension and absorption of data. Computer vision is a method for processing picture input and producing graphic output. To process pictures, computer vision uses object detection, object recognition, object identification, and other techniques. In computer vision applications such as action recognition, car safety, and surveillance, object recognition and tracking are critical.

The use of CNN to detect lane markers, other cars, pedestrians, and bicycles enabled computer vision to attain automobile self-reliance (autonomy vehicle). These systems employ a forward-facing camera. Mobileye has developed unique chips that employ CNNs to detect things in the road ahead. Mobileye was purchased by Intel for $15.3 billion in 2017. Tesla's well-known Autopilot technology uses CNNs to accomplish similar outcomes (I. Vasilev et al., 2019). A convolutional neural network (CNN) is a multilayer neural network modeled after the structure of neurons in the visual brain of an animal (N. Kruger et al., 2013).

Deep convolutional neural networks are a type of computer vision technology that creates a feature hierarchy by layering low-level features to build high-level features. CNN, for example, processes a picture by extracting low-level characteristics from earlier layers, such as edges and smears, which are then merged to generate high-level features, such as object shapes of any type, based on the original item whose feature was extracted and stored in the matrix. In the fields of computer vision and security, CNN is critical. CNN is a feedforward neural network with a number of different special layers. Convolutional layers, for example, apply a filter to the input image (or sound) by sliding the filter across the entire incoming signal, resulting in an n-dimensional activation map.

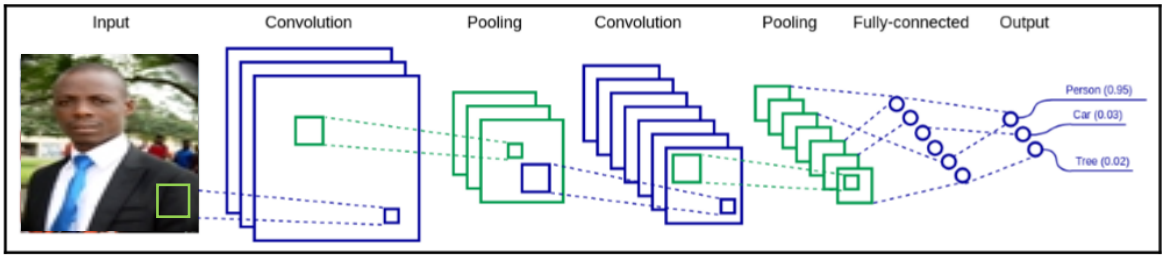


Figure 1: A Basic Convolutional Network Structure with Convolutional and Fully-connected Layers in Blue and Pooling Layers in Green

Figure 1 shows a network structure that employs 3 × 3 convolutions for stride 1 and 2 × 2 pooling for stride 2: The first convolutional layer's neurons accept picture input of 3 × 3 pixels. The first layer's output neurons are merged into a 4 × 4 receptive field size based on the stride. After the first pooling, the set is joined in a single neuron of the pooling layer. The second convolution operation used the input picture to create 3 × 3 pooling neurons and receive input from a square with a side of 3 × 4 = 12 (or a total of 12 × 12 = 144) pixels.

Related Literature

Using Gradient Base Learning to Recognize Documents

The work undertakes document recognition using CNN concept to study and understand neuro-biological stimulation and aggravated by animal visual cortex called the cat visual processing system (Y. LeCun et al. 1998). Convolutional Neural Networks was first introduced in classification detection by Y. LeCun et al., the work undertakes document recognition using CNN concept to study and understand neuro-biological stimulation and aggravated by animal visual cortex called the cat visual processing system.

Using Megapixel Images to Train Convolutional Neural Networks

Average pull convolution was employed in the research of the gradient descent stage in training deep convolutional neural networks. This method focuses on the concept of limited memory to limit the maximum dimension of data input and shows how to train CNN to hold only a portion of the image in memory while still achieving the same results as if all of the data was input. They compared the novel method of training CNN to the classic method and found that the new method produced 97% less memory than the traditional method (H. Pinckaers et al., 2018).

Different Activation Functions for Convolutional Neural Networks

In CNN training, the importance of activation functions is to increase boundary definition, which aids in object categorization, which is the main method to developing an efficient and performing function. The activation function ensures accurate parameter learning and prevents vanishing gradient issues. To improve the performance of CNNs on small/medium size biomedical datasets, a collection of CNNs were trained using multiple distinct activation functions. The results reveal that their suggested ensemble outperforms CNNs that were trained using traditional ReLU as an activation function (H. Pinckaers et al. 2018).

Using Deep Learning to Recognize Facial Expressions

With the advancement of deep learning, the structure of CNNs has become more complex, resulting in improved object recognition performance. The classification mechanism of CNNs, however, remains a major issue. The development and training of a convolution neural network based on facial recognition, as well as the exploration of a neural network classification technique using Deconvolution visualization and a qualitative scheme, show that the trained network express recognition convolution neural network forms a detector for the specific facial action unit. When comparing the maximum distance of all face feature components in the feature graph, the neural system's sensitive feature is dependent on the distance between the system and the object (Yongpei Z. et al., 2019).

Fast Convolutional Neural Network Algorithms

Andrew L., et al.. (2015) attempt to develop a faster deep CNNs algorithm that improved CNNs' performance in object recognition. CNNs' image recognition success in self-driving cars and mobile phones is constrained by processing resources. Andrew describes a new family of CNN algorithms that use Winograd's minimum filtering techniques and replace the GPU with a VGG (Visual Geometry Group) network, demonstrating state-of-the-art throughput for batch sizes ranging from 1 to 64.

Framework for Concatenating Shortcut Convolutional Neural Networks

The importance of CNNs in learning image features for classification and recognition, as well as the limitation that CNNs only allow connections between adjacent layers, limiting multi-scale data integration. The application of a concatenation framework of shortcut CNNs is an approach that may improve CNNs. By using shortcut connections to the fully-connected layer, which is directly fed to the output layer, the framework concatenates multi-scale features.

Yujian Li et al. (2017) conducted numerous experiments using various visual datasets, including AR, FERET, FaceScrub, CelebA for gender classification, CUReT for texture classification, MNIST for digit recognition, and CIFAR-10 for object recognition, for various tasks in order to determine the performance of the shortcut CNNs. On the tasks, the shortcut convolutional neural networks outperform typical conventional CNNs, with greater stability over a range of pooling schemes, activation functions, optimizations, initializations, kernel numbers, and kernel sizes.

Convolutional Neural Network Extension Using General Image Processing Kernels

The use of a filter (or pre-defined kernels) for image processing in CNN, which denies CNN the ability to find its kernels by using 41 different general-purpose kernels of blurring, edge detection, sharpening, discrete cosine transformation, and so on in the first layer of the CNNs, was designed to reduce CNN training time. The architecture, which was termed General Filter Convolutional Neural Network, was able to reduce training time by 30% while maintaining good accuracy when compared to typical CNNs (GFNN). It was also revealed that GFNN can be taught to achieve 90% accuracy with only 500 samples, albeit this was not for the MNIST dataset (J. H. Jung et al., 2019).

Used of Convolutional Neural Networks to create a Face Recognition Library

The face recognition library was created with the goal of making face recognition features easier to integrate in mainstream applications. CNNs were used to construct the library. It describes the overall

library structure and assesses it in a larger-scale scenario including a pre-trained image. When utilizing the required confidence of 90%, this technique attained an accuracy of 98.14% with the recommended library, and 99.86% otherwise (Leonardo B., & Alison R. P. 2017).

Methodology

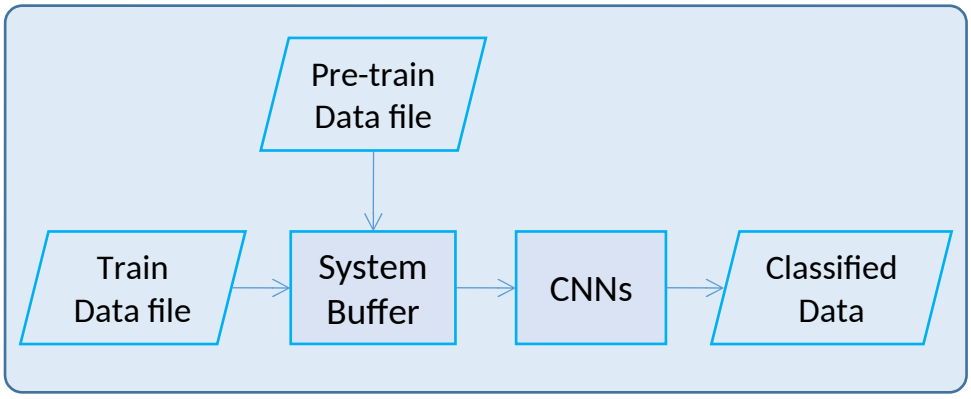


Figure 2: Conceptual Framework

The research conceptual framework is depicted in Figure 2. The framework was given two sets of data to deal with: the pre-train and training datasets. These datasets are saved in the system buffer and fed through a convolutional neural network for processing, with the classified data being the ultimate result obtained as the system's prediction.

The symbolic property of an existing network, which is the output of the adopted net's last convolutional (or pooling) layer, is the starting point for transfer learning. The pooling layer's features are converted into a new set of classes for the new challenge. This is accomplished by replacing the existing pre-trained network's last completely linked layer with a new layer that represents the classes. The pooling layer's features are converted into a new set of classes for the new challenge. This is accomplished by replacing the last fully-connected layer of the old pre-trained network with a new layer that represents the new problem's classes, and then training the new layer with a dataset relevant to the new job as in Figure 3.

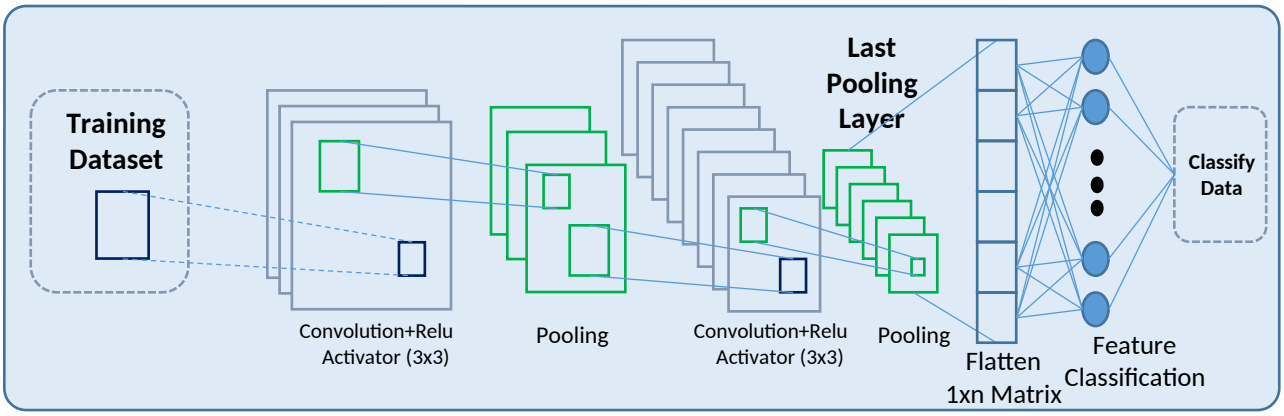


Figure 3: CNN Transfer Learning Model for Pre-trained Network

The input to the system model is an image, and the output is a label for each object in the image, as well as the probability of accuracy for each detected object category.

Figure 3 depicts the system CNN model, which includes two learning and classification features. However, before the pooling stage, our model employed the ReLu activation function in the convolution layers. Following the first stage, the second stage is carried out by using the output of the previous layer as input data. A total of 144 layers are used in the procedure. To reach the ultimate result, which is the network prediction, the classification feature is flattened with a full connection.

Data Analysis

Only 175 images are loaded into the buffer in the training dataset, which is referred to as new images. We separate the data into training and validation sets. 70% was used in the training set, whereas 30% was used in the validation set. GoogleNet Network provided the pre-trained dataset, which was examined using the Deep Learning Toolbox Model for GoogLeNet Network support package.

The picture input layer is the first element of the network's Layers property, as seen in Figure 3. The layer allows photos with dimensions of 224 × 224 × 3, with 3 being the number of color channels, as defined by Google Net guidelines.

Replacement of Layers

The last layer of the old network had to be updated in order to retrain a pre-trained network to classify fresh photos. We replace these two layers with new layers that are tailored to the new dataset, and we verify that the new layers are connected appropriately by projecting the new layer graph on the network's final layers as in Figure 4.

The network's convolutional layers extract visual attributes that the final training layer and classification layer utilize to classify the input image. In GoogLeNet, these two layers are loss3-classifier and output. They contain instructions on how to combine the characteristics extracted by the network into class probabilities, loss values, and predicted labels.

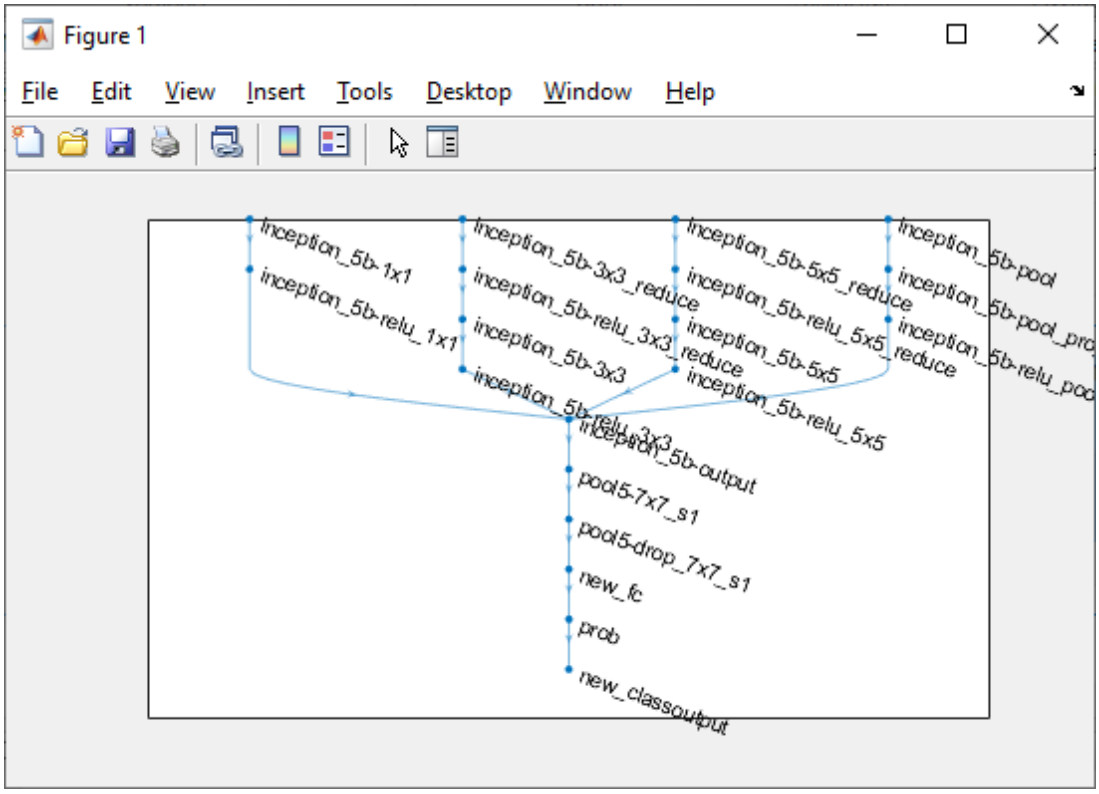


Figure 4: Confirmation of New Layers Connected Correctly

Train Network

We retrained our network on the new set of images at this point. Because our dataset is limited, we set the learning rate to zero to freeze the weights of the previous layers in the network to prevent overfitting.

The network's input images are 224 × 224 × 3 pixels in size. Because the photos in our dataset aren't all the same size, we enrich the images dataset to avoid the network from overfitting and memorizing the training images' exact characteristics.

We define the training parameters, including the number of epochs to train for, the mini-batch size, and the validation data.

Computation Accuracy

As illustrated in Figure 5, we compute the validation accuracy per epoch. We use the fine-tuned network to classify the validation images and calculate the classification accuracy, which averages 95%.

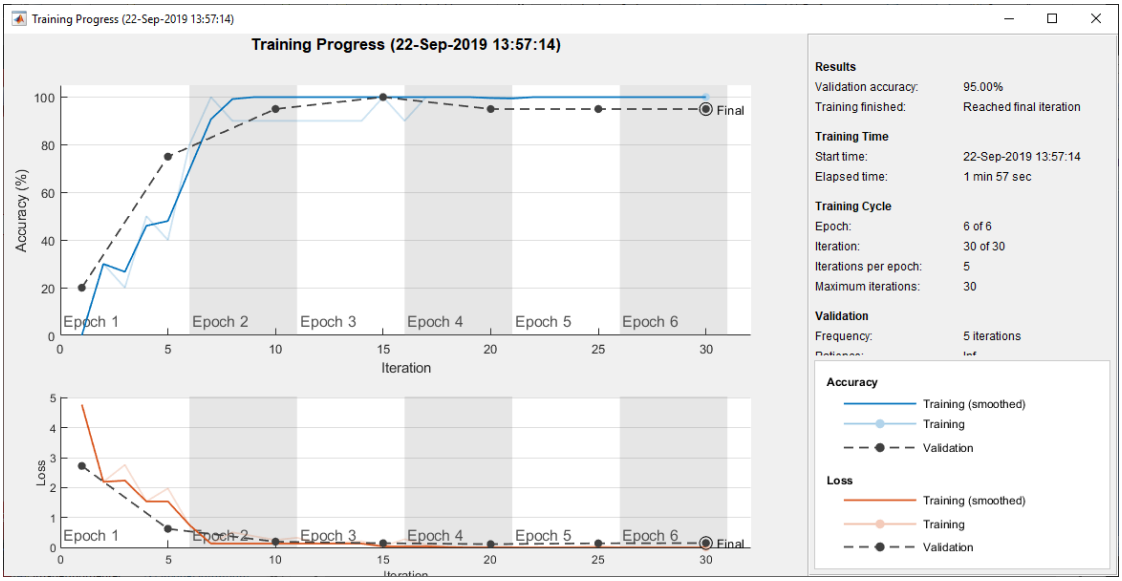
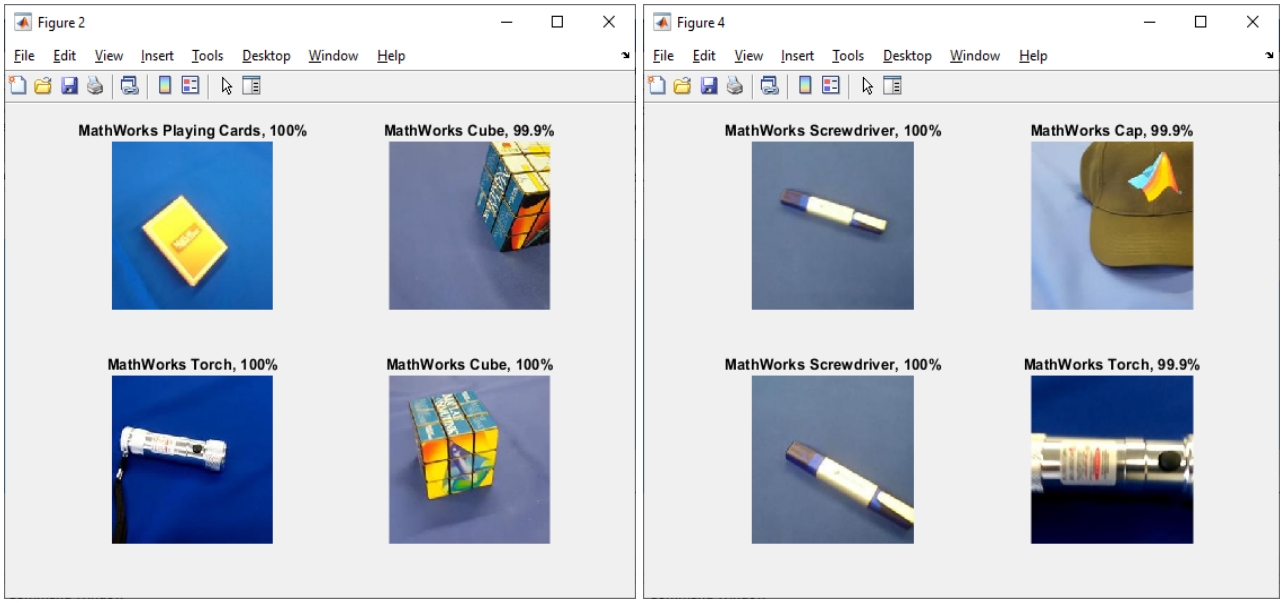
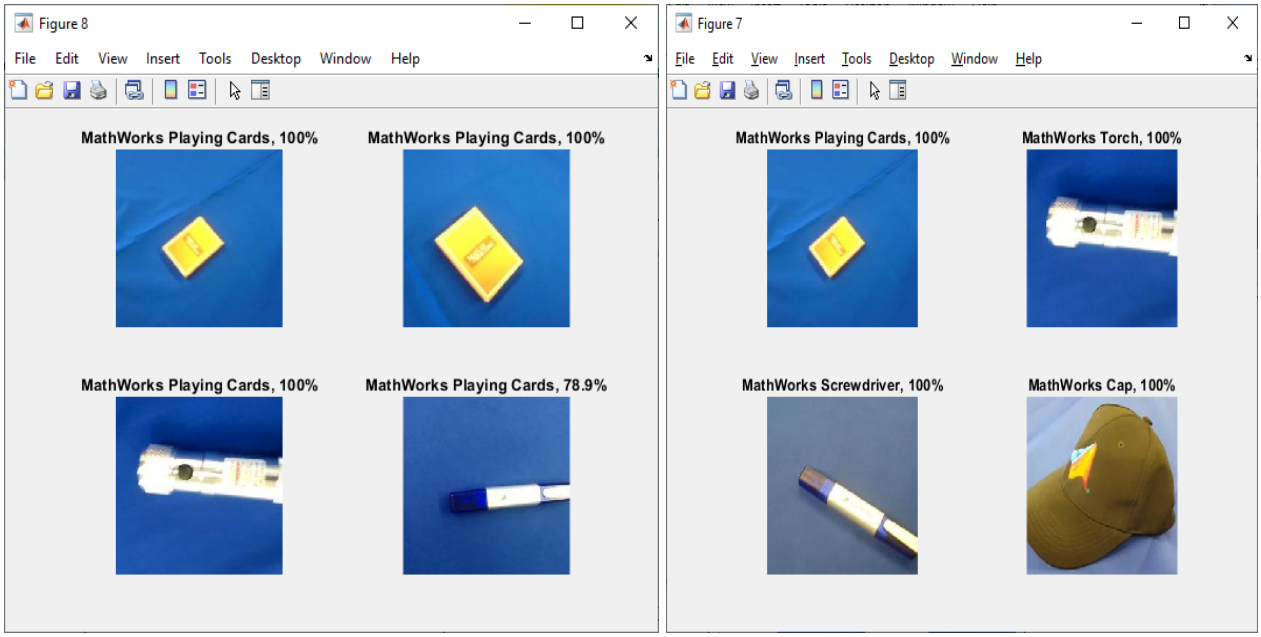


Figure 5: Training Accuracy and Loss

System Visual Prediction

Figure 6 is a visual presentation of photos of typical commodities taken from a grocery area, and it shows how the algorithm identified them. Through the CNN method's training and classification procedure, the system was able to classify each image. The method suggests an image based on the name and the accuracy percentage.





Conclusion

Transfer training minimizes the number of epochs and layers required to train a dataset and speeds up CNN training time. It also makes training tiny datasets without overfitting much easier. This study of image classification utilizing the CNN transfer training technique yielded 95% image prediction accuracy on a new dataset of 75 photos.

References

1. Andrew L., Scott G. (2015). Fast Algorithms for Convolutional Neural Network. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4013-4021, https://doi.org/10.1109/CVPR.2016.435
2. BVLC GoogLeNet Model. https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet
3. I. Vasilev, D. Slater, Gianmario S., P. Roelants, V. Zocca. (2019). Python Deep Learning Second Edition, Packt Publishing
4. J. H. Jung, Yousun S., Young M. K. Extension of Convolutional Neural Network with General Image Processing Kernels. TENCON 2018 - 2018 IEEE Region 10 Conference, 1436-1439. https://doi.org/10.1109/TENCON.2018.8650542
5. Y. Lecun, L. Bottou, Y. Bengio, P. Haffner. (1998). Gradient Based Learning Applied to Document Recognition. Proceedings of the IEEE, 86 (11), 2278–2324. https://doi.org/10.1109/5.726791
6. Leon A. Gatys, A. S. Ecker, Matthias B. (2015). A Neural Algorithm of Artistic Style. https://arxiv.org/abs/1508.06576
7. Leonardo B., Alison R. P. (2017). A Face Recognition Library using Convolutional Neural Networks. International Journal of Engineering Research and Science (IJOER), 3 (8), 84-92
8. N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. J. Rodriguez-Sanchez, L. Wiskott. (2013). Deep hierarchies in the primate visual cortex: What can we learn for computer vision? Pattern Analysis and Machine Intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (8), 1847-1871 https://doi.org/10.1109/TPAMI.2012.272
9. H. Pinckaers, G. Litjens. (2018). Training convolutional neural networks with megapixel images. 1st Conference on Medical Imaging with Deep Learning (MIDL)

10. Yongpei Z., Hongwei F., Kehong Y. (2019). Facial Expression Recognition Research Based on Deep Learning. https://doi.org/10.48550/arXiv.1904.09737
11. Yujian L., Ting Z., Zhaoying L., Haihe H. (2017). A concatenating framework of shortcut convolutional neural networks. https://doi.org/10.48550/arXiv.1710.00974