wjert, 2022, Vol. 8, Issue 4, 285-302. *Review Article* ISSN 2454-695X

World Journal of Engineering Research and Technology

WJERT

www.wjert.org SJIF Impact Factor: 5.924

WILDLIFE CLASSIFICATION MODEL – DEEP LEARNING IN PYTHON WITH KERAS

Mohammed Zidan Vaheed*[1] and Dr. M. N. Nachappa[2]

[1]Post Graduate Student, Department of Master of Computer Applications School of CS & IT JAIN(Deemed-to-be-University) Bangalore, India.

[2]Professor and Head, School of Computer Science and Information Technology, JAIN (Deemed to be University) Bangalore, India.

Article Received on 26/01/2022 **Article Revised on 16/02/2022** **Article Accepted on 06/03/2022**

***Corresponding Author**
Mohammed Zidan Vaheed
Post Graduate Student,
Department of Master of
Computer Applications
School of CS & IT JAIN
(Deemed-to-be-University)
Bangalore, India.

ABSTRACT

Deep learning algorithms are a subset of the machine learning algorithms, which aim at discovering multiple levels of distributed representations. Recently, numerous deep learning algorithms have been proposed to solve traditional artificial intelligence problems. This work aims to create a state-of-the-art deep learning model with computer vision at its base by highlighting the contributions and challenges from recent research papers. It first goes over the recent studies on the deep learning topics and the discoveries of other brilliant individuals in this field and highlights their successes and how that has contributed to the creation of our model. The system requirements are elaborated to help understand the intensity of work the system has to handle and what minimum level of computational power is required to create these deep learning models, as we know the models are quite computationally power heavy and need a lot of resources and time to work and create these models. Then we see the overview of the system, analyzing it as well as its designs using pictorial representations to make understanding of the topic much easier. Finally, we summarize the discussion with the future plans for this deep learning model and how it will be upgraded for obtaining greater results that far surpass the results we have obtained from this project now.

KEYWORDS: Artificial Intelligence, Computer vision, Power heavy, Convolution Neural Networks, Image Recognition.

INTRODUCTION

Deep learning is a subfield of machine learning which attempts to learn high-level abstractions in data by utilizing hierarchical architectures. It is an emerging approach and has been widely applied in traditional artificial intelligence domains, such as semantic parsing, transfer learning, natural language processing, computer vision and many more. There are mainly three important reasons for the booming of deep learning today: the dramatically increased chip processing abilities (e.g., GPU units), the significantly lowered cost of computing hardware, and the considerable advances in the machine learning algorithms. Deep networks have been shown to be successful for computer vision tasks because they can extract appropriate features while jointly performing discrimination. In recent ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competitions, deep learning methods have been widely adopted by different researchers and achieved top accuracy scores.

The current image recognition methods use artificial extraction features. This method is not only time-consuming and laborious, but also difficult to extract. However, deep learning is a kind of unsupervised learning. In the process of learning, it is not necessary to know the tag value of the sample, and the whole process can extract good features without human participation. In the recent years deep learning has become a hot topic of research in the field of image recognition. It has achieved good results and has a broad space for research.

The most famous aspect of computer vision deep learning models is the use of Convolution Neural Networks (CNN). The convolutional neural networks (CNNs) can use the model structure of the convolution layer and the lower sampling layer in turn by simulating the human visual system. The convolution layer enhances the original signal and improves the signal to noise ratio. The lower sampling layer uses the principle of image local correlation to sample the image from the neighborhood. It can extract useful information while reducing the amount of data. At the same time, parameter reduction and weight sharing alleviate the problem of long training time to a certain extent. However, experiments show that the training time of CNNs is still very long.

In this paper we aim to create convolution neural networks that can recognize the images of animals and accurately classify them. Alongside this we aim to reduce the time of training for the CNNs which are famously slow as well as add extra layers into the neural networks to get greater results for our deep learning model.

Applications

Any form of identification system will always have relevance in today's world, be it a human recognition system or even an animal recognition system. These systems can be applied in various fields or research, study, or identification in real time. However, this is a growing field and like every growing field there will be new technology, new software's, new methods to implement existing projects so that it can be improved and there is no doubt that the same can be said for the field of computer vision i.e., image recognition and classification.

A few applications for a system that can identify wildlife animals are listed below

- **Identification of endangered species:** Certain wildlife organizations and conservation groups can make use of such systems to identify endangered species in the wild and take measures to protect them from poachers and hunters to ensure the survival of their species in the current world.
- **Research and Study:** Researchers can make use of systems to understand how the system identifies the animals and what specific features help distinguish certain types of animals that look the same but are not the same, such as a horse and a mule or a deer and an antelope.
- **Disease identification:** If the systems were to be advanced in any way, the best path to take would be to implement a disease recognition aspect into the system. Human and animal bodies are different and the way diseases affect them are different as well. To be able to study diseases and identify them in an animal's body will help in medical treatments for animals as well as it will help researches study what a certain disease looks like when it affects a certain species of animals.

Available Datsets

Datasets for animals exist in a large variety and amount. To get the right dataset it is quite difficult as a lot of the animal datasets that exist on Kaggle are not up-to the expected level that is required to train a model for image recognition. Some of the useful datasets that can be used are

- **Animals-10:** This dataset was created by Corrado Alessio for their matriculation thesis. It consists of 28 thousand medium quality animal images belonging to 10 categories which are: Dogs, Cats, Horse, Spiders, Butterflies, Chickens, Sheep, Cows, Squirrels and Elephants. All the images have been collected from Google Images and have been checked properly by human hands. There are some erroneous data to simulate real conditions.
- **African Wildlife:** This dataset was created by Bianca Ferreira with the original goal of training an embedded device to perform real-time animal detection in nature reserves of South Africa. This dataset was also crafted using images from Google Images. Compared to the other datasets this one lacks in categories but it has a considerable number of images that can be used to create image recognition models.
- **Oregon Wildlife:** This dataset was uploaded on Kaggle by David Molina. He discovered the dataset using a google scrapper from GitHub. Forked the repository, started a new branch and performed some adaptations of the code and downloaded the data set. This dataset consists of 20 categories of animals. It can be of great use to train stronger image classification models with its high variety of categories.
- **STL-10 Image Recognition Dataset:** STL-10 is an image recognition dataset inspired by CIFAR-10 dataset with some improvements. With a corpus of 100,000 unlabeled images and 500 training images, this dataset is best for developing unsupervised feature learning, deep learning, self-taught learning algorithms. Unlike CIFAR-10, the dataset has a higher resolution which makes it a challenging benchmark for developing more scalable unsupervised learning methods.
- **Animals Detection Images Dataset:** An animals detection dataset that was extracted using Google Open Images V6+. Consists of 21 categories of animals each containing 50+ pictures of each species of animals.

Related Work

The use of neural networks to a variety of computer vision subjects has demonstrated that with the correct resources, computer vision can recognize almost everything. Obtaining the required datasets for the project is arguably the most difficult component of a process like this. The dataset will serve as the project's foundation, since neural networks will use it to train and learn how to recognize a specific image. The effort of figuring out and putting up networks, as well as strategies to improve them, will be easier to figure out and implement with the aid of other publications; datasets are irrelevant. What counts are the networks that have been built, the outcomes that have been produced, and how they were developed. In recent literatures there have been several approaches to using deep convolution neural networks for image classification, but since convolution neural networks are more scalable for larger datasets it is more suitable to apply them for our wildlife classification as the datasets being used are enormous.

In[1] they used convolution neural networks to classify food images, to be more precise they used an inception v3 model which was pre-trained by ImageNet. The convolutional neural network learns the filters that were previously hand-engineered in existing methodologies. They conducted their investigation using the Food-11 Dataset, which included 16643 photos divided into 11 categories. Dataset sizes are often in this range, which is one of the key reasons why training neural networks takes so long.

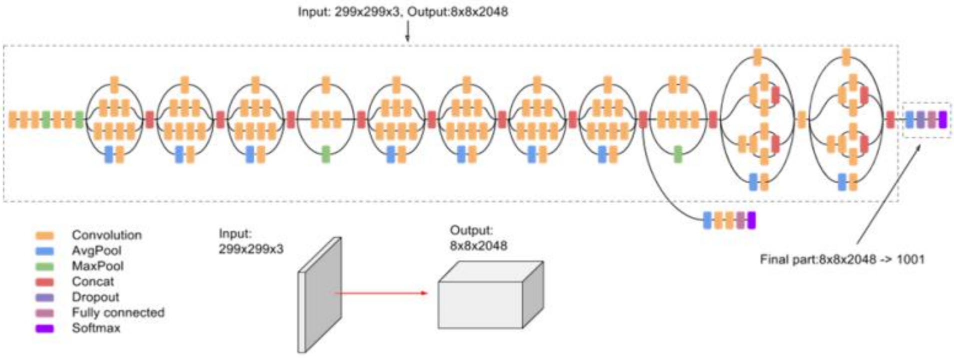


Fig. 1: Architecture of the inception v3 model.[1]

The inception v3 model consists of the following layers

- **Convolution Layer:** At the beginning convolution layer with input size 299 x 299 x 3 to create feature maps by convolving input images.
- **Max Pooling Layer:** Max-pooling is a sample-based discretization process. Max pooling is done by applying a max filter to non-overlapping sub regions of the input matrices. Max-pooling extracts the most important features like vertical edges and horizontal edges.
- **Average Pooling Layer:** Average pooling layer reduces the variance and complexity in the data. It also divides the input into rectangular pooling regions and computing the average values of each matrix to down sample the input features.[10]
- **Concat Layer:** The Concat layer concatenates its multiple input blobs to one single output blob.[10]

- **Dropout Layer:** The dropout layer randomly drops elements from a layer in the neural network. Dropout is a technique used to improve over-fit on neural networks.
- **Fully Connected Layer:** The fully connected (FC) layer in the CNN represents the feature vector for the input. This feature vector holds information that is vital to the input.
- **SoftMax Layer:** The SoftMax assigns decimal probabilities to each class in a multi-class recognition problem. Those decimal probabilities must add up to 1.0. This additional constraint make training to converge more quickly.

In[2] the researches had opted to use the deep belief networks and they had studied the structure of the restricted Boltzmann machines (RBM). They opted to go for a SoftMax classifier for their deep belief network and carried out their experiment on the MNIST library. Their experiment showcased that the recognition rate of the deep belief network was basically flat and slightly decreased, but the training time was greatly shortened. They concluded by mentioning that the combination of random regression and drop sampling obviously improves the recognition rate of the deep belief network system and reduces its training time. The training methods used by DBNs are also very different from those of traditional neural networks. For the three-layer network, the BP algorithm has a good training time and algorithm recognition rate. However, for a network with multiple hidden layers, the training time is too long.

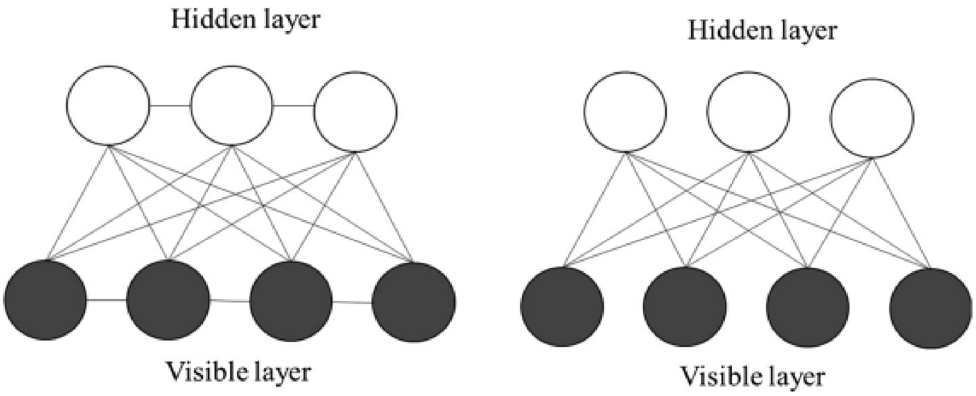


Fig. 2: BM and RBM.[2]

The training process of DBNs is trained by layer by layer, and only one layer of RBM is trained at each time. This process is exactly the same as the RBM training, and the parameters are adjusted separately.

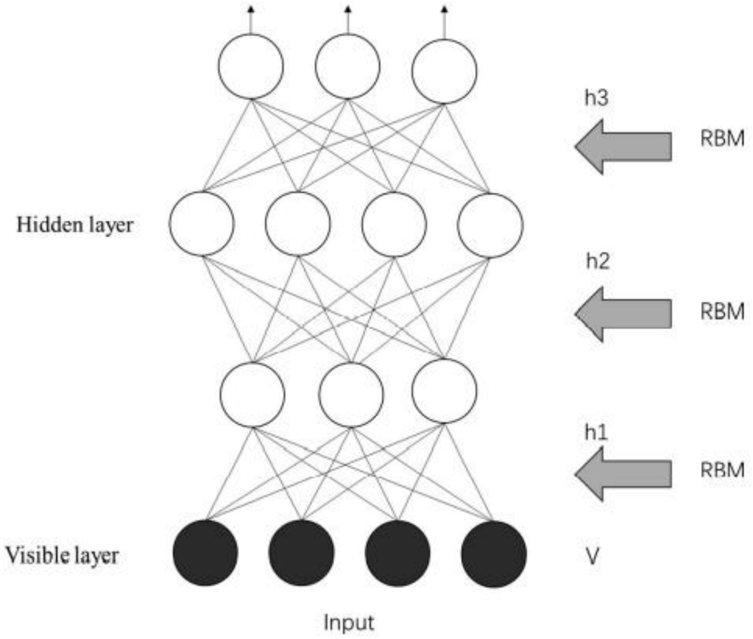


Fig. 3: Structure of Deep Belief Network.[2]

It can be divided into two parts

- Step 1 - Pre-training First, the parameters of the deep belief network are set up. The number of nodes on each tier, as well as other aspects of the DBNs network, such as Dropout size, sparsity, and noise, are all part of the initialization process. After startup, each layer's RBM may be trained independently. The first level's output h1 becomes the second level's input, and so on, with the weight W_{ij} of each tier being preserved.
- Step 2 - Fine tune in order to improve the performance of the network, the whole network may be changed based on the sample's label value. It is decided to employ the gradient down algorithm. The DBNs network is now an ordinary neural network, comparable to the BP algorithm. The weights of all layers are trained in advance before fine-tuning. Because it is not randomized like a neural network, it just requires a limited number of iterations to achieve decent results.

In[3] the researches have reviewed an interesting topic in the computer vision field of image recognition and classification with the neural networks which is the implementation of Semantic Segmentation by combining Conditional Random Fields (CRFs) and Deep Neural Networks (DNNs) as both of these topics have shown excellent results in their fields. The idea of Semantic Segmentation was to label every pixel in an image with a pre-defined object category. It has numerous applications in scenarios where the detailed understanding of an image is required, such as in autonomous vehicles and medical diagnosis.

Picture classification, for example, provides a high-level description of the image by categorizing whether or not particular tags exist. Object detection, semantic segmentation, and instance segmentation are some of the other activities that give more comprehensive and localized information about the scene. With tasks like picture captioning and visual question-answering, researchers have begun to bridge the gap between natural language processing and computer vision.

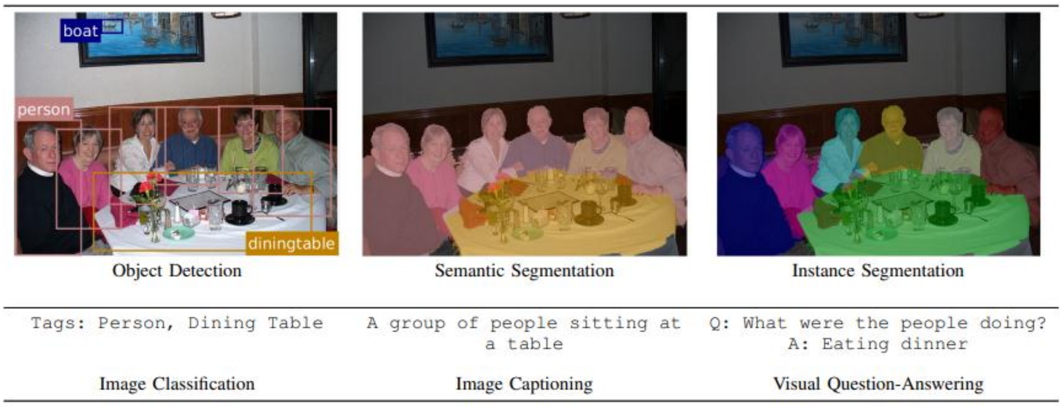
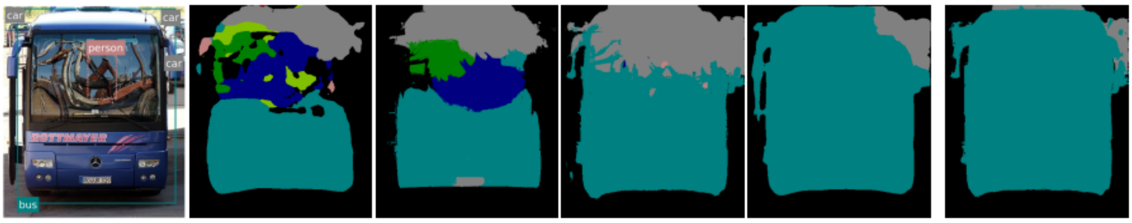


Fig. 4: Example of various scenes under different segments.[3]

Multiple strategies can be employed in semantic segmentation to obtain the required outputs with the maximum possible accuracy. FCN has a tendency to create "blobby" outputs that ignore the image's edges (from the Pascal VOC validation set). DeepLab, which uses DenseCRF to refine the outputs of a fully-convolutional network, delivers a result that is compatible with the image's edges. CRF-RNN and DPN both train a CRF simultaneously within a neural network, outperforming DeepLab. Unlike other methods, the Higher Order CRF can recover from incorrect segmentation unaries since it also uses cues from an external object detector, while being robust to false-positive detections (like the incorrect "person" detection). Object detections produced by have been overlaid on the input image, but only uses this information.



Input FCN DeepLab CRF-as-RNN DPN Higher Order CRF

Fig. 5: Comparison of various Semantic Segmentation.[3]

In[4] the idea of convolution neural networks is discussed but with the idea of improving the accuracy along with trying to overcome some of the issues such as shallow number of layers as well as the issue of over fitting. They use the cat and dog's dataset for this experiment while implementing the VGG16 model of the convolution neural networks with adjustments of their own. The experimental results showed that the improved model can greatly improve the detection accuracy.

The VGG16 network VGG16 was proposed by the VGG (Visual Geometry Group, VGG) of Oxford University. It was the basic network in the 2014 ImageNet competition positioning task first place and the second-place classification task. Observing the previous AlexNet and XF-Net, we can find that the convolution kernel is getting smaller and the network is deepening, which is conducive to improving accuracy. As a result, VGG16 has a deeper network and more efficient resource use than the previous one. The recognition effects of the models are higher.

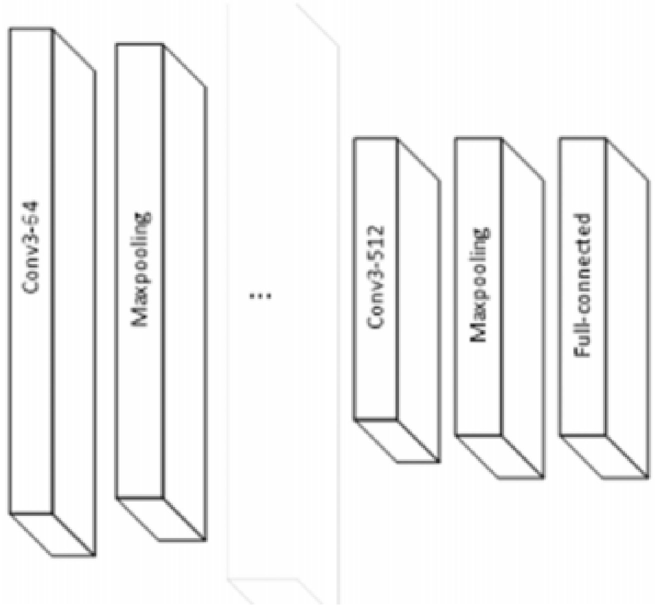


Fig. 6: VGG16 Network Model Diagram.[4]

This paper combines the VGG16 network and the feature fusion layer built by.[4] to get a new network. In this paper, a feature fusion layer is designed, which can perform non-linear combination of the feature information learned by the VGG16 network, so as to obtain more non-linear combination results. So as to achieve better results.

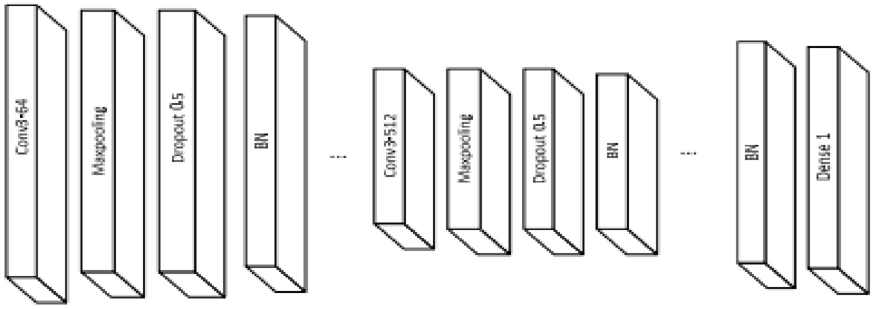


Fig. 7: Improved model of VGG16.[4]

In[5] the paper aimed to review the state-of-the-art deep learning algorithms in computer vision by highlighting the contributions and challenges from recent research papers. It first gives an overview of various deep learning approaches and their recent developments, and then briefly describes their applications in diverse vision tasks. It covers topics such as CNN layers, training strategies, Restricted Boltzmann Machines (RBM's), Deep Belief Networks (DBN's), Deep Energy Models (DEM's) and their respective applications. Finally, the paper summarizes the future trends and challenges in designing and training deep neural networks.

The research done here[6] aims to present a scalable system capable of examining images and accurately classifying the image based on its visual content. When retrieving images based on a user's query, the system will yield a minimal amount of irrelevant information (high precision) and insure a maximum amount of relevant information (high recall). The dataset been used for identification and classification here is for sports.

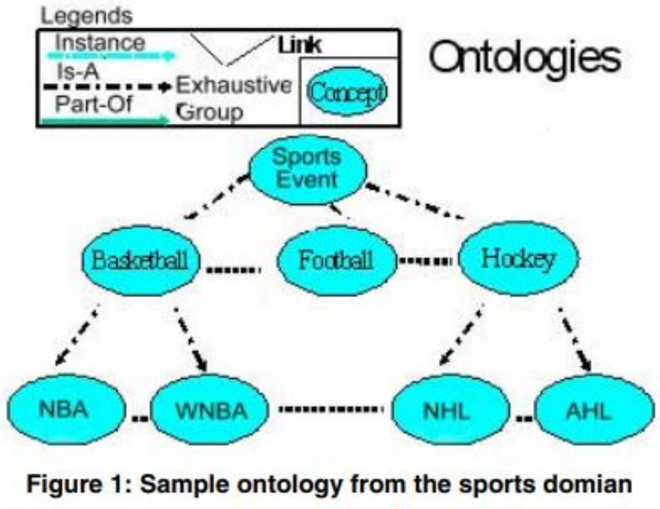


Fig. 8: Sample ontology from the sports domain.[6]

An ontology is a specification of an abstract, simplified view of the world that we wish to represent for some purpose. As a result, an ontology specifies a collection of ideas, which are representational words. A target world is described by the interrelationships between these notions. There are two techniques to build an ontology: domain-specific and general. Generic ontologies include CYC and WordNet. They.[5] chose to use a domain-dependant ontology. A domain-dependent ontology gives fine-grained concepts, whereas generic ontologies provide coarser-grained concepts. Fine-grained concepts allow us to identify particular links between aspects in pictures that may be utilised to categorise them successfully.

In this paper,[7] they have applied deep learning to create a handwritten character recognition system, and explored the two-mainstream algorithm of deep learning: the Convolutional Neural Network (CNN) and the Deep Belief Network (DBN). They conducted the performance evaluation for CNN and DBN on the MNIST database and the real-world handwritten character database. From the experiments the primary differences between the DBN and CNN became quite clear, which were

- DBN is a generation deep model that belongs to the unsupervised learning technique, whereas CNN is a discrimination deep model that belongs to the supervised learning method.
- DBN is superior for one-dimensional data modelling, such as speech, but CNN is better for two-dimensional data modelling, such as pictures.
- CNN is basically an input-output map. While DBN must construct a joint probability distribution between visible and hidden units, as well as the marginal probability distribution of visible and hidden units, it can learn a large number of mapping relations and does not require any specific mathematical expressions.

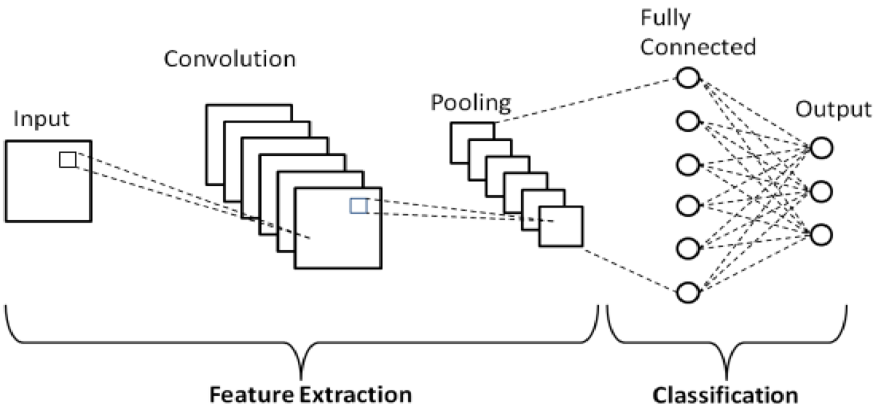
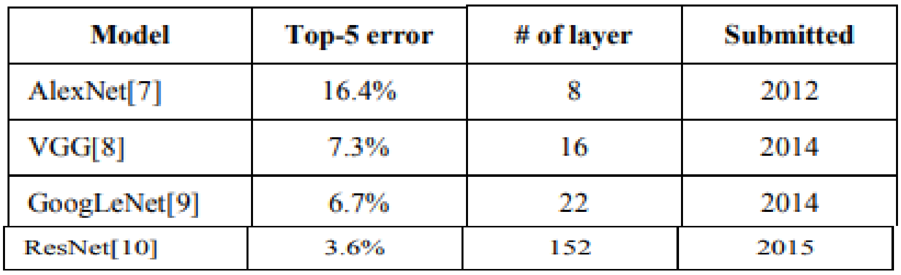


Fig. 9: Simple Structure of CNN.[7]

CNN and DBN had classification accuracy rates of 99.28 percent and 98.12 percent on the MNIST database, respectively, and 92.91 percent and 91.66 percent on the real-world handwritten character database.

This study.[8] tries to classify the leaves affected by several types of pest attacks automatically. Using the well-known VGG16 architecture, Convolutional Neural Network (CNN) is one of the most recent classification methods reported in this study. When weights are randomly initialised, VGG16 training can last a long period. As a result, they used transfer learning to pick beginning weights in order to enhance accuracy and reduce training time. To acquire the best results, several situations are examined using a combination of the number of learnable parameters and optimizer types. ImageNet data with 1.2 million colour photos and 1000 classes was used to train the VGG16 model. All kernel sizes are 3x3 in the original VGG16, which contains 16 convolution layers using the ReLU activation function. After each convolution layer, a max-pooling layer with all 2x2 kernel sizes follows. Convolution layers are used to retain training weights and operate as an automated feature extraction system. The next layer is 3 fully connected layers (FC) which are the final layer as a classifier. The weight of the training results can be stored by the convolution layer and FC, allowing them to calculate the number of parameters. Because most of the parameters were convergent, transfer learning normally utilises a learning rate with relatively modest values. In most cases, the number of epochs is not excessive. The number of epochs used is 100 with a learning rate of 0.0001. The comparisons made here were using SGD and the Adam optimizers. The results obtained are in favour of the SGD optimizer.

This paper[9] gives a brief explanation into what the different types of approaches to deep learning are such as, AlexNet, Visual Geometry Group Model, GoogLeNet and ResNet. The comparison between these topics is summarized in a table that describes each of the models with the layers they use and the loss percentage over the years.



Model	Top-5 error	# of layer	Submitted
AlexNet[7]	16.4%	8	2012
VGG[8]	7.3%	16	2014
GoogLeNet[9]	6.7%	22	2014
ResNet[10]	3.6%	152	2015

Fig. 10: Final Summary of the 4 models.[9]

Since deep learning has been progressing a lot in the past years this paper serves as a way of reviewing the relevant studies related to deep learning.

This paper[10] proposed a deep convolutional neural network architecture codenamed Inception that achieves the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The enhanced usage of computer resources inside the network is the fundamental feature of this design. It was possible to enhance the network's depth and width while keeping the computational budget same by carefully constructing this architecture. The architectural selections were based on the Hebbian principle and the understanding of multi-scale processing to enhance quality. The Inception architecture began as a case study for evaluating the putative output of a complex network topology design method that attempts to simulate a sparse structure for visual networks by using dense, easily available components to cover the expected outcome. With a bit of optimization, the Inception model proved to be especially useful in the context of localization and object detection. The basic idea behind the Inception architecture is to think about how a convolutional vision network's optimal local sparse structure can be approximated and covered by readily accessible dense components. The architecture of the Inception model had 2 variations, one known as the naïve version and the other was dimensionality reduced version. The naïve versions incarnations were restricted to 1x1, 3x3 and 5x5 filter sizes as not doing so would lead to patch-alignment issues in the model. As these “Inception modules” are stacked on top of each other, their output correlation statistics are bound to vary: as features of higher abstraction are captured by higher layers, their spatial concentration is expected to decrease. This suggests that the ratio of 3×3 and 5×5 convolutions should increase as we move to higher layers.

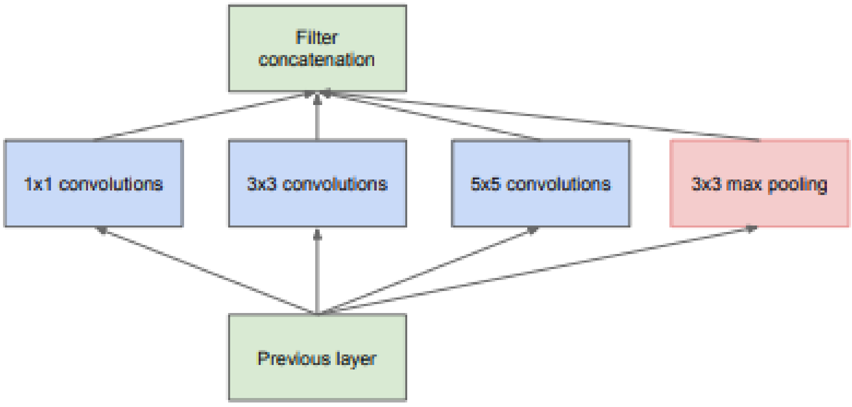


Fig. 11: Inception module, naïve version.[10]

However, on top of a convolutional layer with a high number of filters, the 5x5 convolutions might be too costly in this model. When pooling units are added to the mix, the number of output filters equals the number of filters in the previous step, making the situation even worse. When the output of the pooling layer is combined with the outputs of the convolutional layers, the number of outputs will always rise from stage to stage.

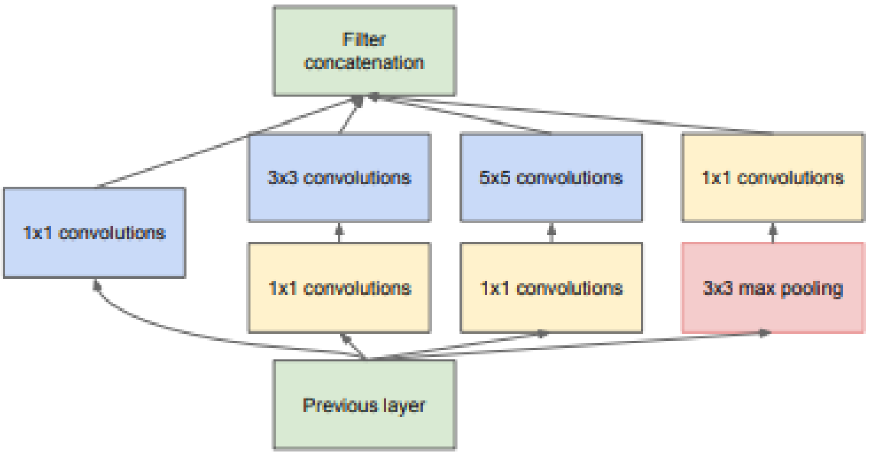


Fig. 12: Inception module with dimensionality reduction.[10]

This leads to the second variation, the inception module with dimensionality reduction. With the help of this the dimensions are reduced significantly where computational requirements increase drastically. This avoids the computational blow up that would eventually occur in the ever-increasing naïve version of the Inception module.

The convolutional neural network VGG16 model is used to address the issue of identifying and classifying household rubbish in this research,[11] which investigates the use of deep learning in the field of environmental protection. This approach first locates and selects the recognised items using the OpenCV computer vision library, then pre-processes the photos into 224×224-pixel RGB images that the VGG16 network accepts. After that, a VGG16 convolutional neural network based on the TensorFlow framework is developed, with the RELU activation function and a Batch Normalization layer added to speed up the model's convergence while maintaining recognition accuracy.

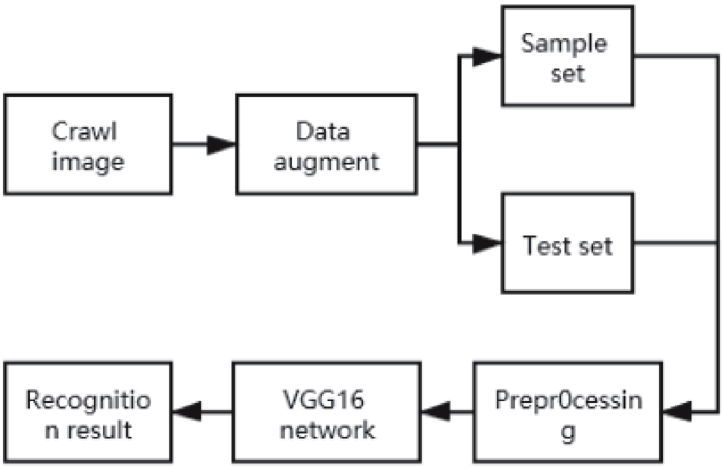


Fig. 13: Overall System Flowchart.[11]

The inserted RGB three-channel image is converted into a single-channel grayscale image with a pixel value of 0 to 255 using the Cvtcolor function. The Gaussian blur operation is performed using the Gaussian blur function on the grayscale image to obtain the new image. Each pixel of the grayscale image is convolved to extract important features of the image to weaken reflective points and noise. The image is then converted into black and white using the Threshold function by performing Global Threshold Binarization. The contour list and its index are obtained after using FindContours function, the bounding rectangle of the contour is obtained according to the BoundingRect function. Finally, the image is rescaled into an RGB image of 224x224 pixels.

This paper.[12] studies the problem of learning good feature in an unsupervised way to obtain better performance in image recognition. Deep network characteristics are supposed to be more dependable and capable of providing more semantic information. Low-level layers, in particular, identify basic characteristics that are passed into higher layers so that the abstract representation in the high levels can resolve lower-level picture ambiguities or infer the positions of concealed object pieces. For encoding, it employs neural networks, polynomial kernel SVM, autoencoders, the Gaussian Mixture Model, and the fisher vector.

In this work,[13] the authors make an effort to deal with the challenges (small size of the objects, imbalanced data distributions, limited labelled data samples) through a computational framework by incorporating latest developments in deep learning. The suggested approach used a two-stage detection scheme, pseudo labelling, data augmentation, cross-validation, and ensemble learning to obtain better results for practical image recognition applications than the typical deep learning methods. The suggested framework was also used as the key kernel in various Kaggle image recognition competitions. The paper tackles two main issues regarding detection of small objects in large images such as real-world images and dealing with imbalanced datasets, which are the most common types of datasets to work with. Regarding the detection of small objects, a method proposed by the authors was to deploy a two-stage detection scheme to cater for small object recognition. This scheme made good use of the advantages of the object detection methods and image classification methods such as ResNet and Faster R-CNN. Regarding the imbalanced datasets the authors suggested the use of data augmentation and cross validation that can help to increase accuracy of the classifier. Their methods consisted of oversampling the rare samples by horizontal flipping, slight shifting and rotation, and oversampling the rare samples by changing colour channel variance. They also split the training dataset into different folds during cross validation so as each fold contained enough so that the classes are represented equally in the dataset. Using this they were able to reduce the negative effect of data imbalance to a minimum.

Challenges

- Datasets are difficult to obtain. Especially good ones that possess all the resources you would need. Some datasets will have less than 5 categories while others will have lesser images to use to train models. The larger datasets are harder to find but they also have the issue of working with large volumes of data which can pose issues for people who have restrictions on data usage.
- Time is a resource that takes a heavy load in model training. Most people try to achieve better models with the hope that they will have lesser training time to make the most out of the time being used to train the models. Dealing with the time consumption is a massive challenge for training models which use large datasets.
- Language barriers can pose an issue when using datasets as some of the datasets are not labelled in English but rather, they are labelled in the creator’s mother tongue. So having to create our own English labels for the dataset will be time consuming.
- Hyperparameter tuning is a skill that is hard to achieve due to the fact that tuning a model takes a long time to perfect. It requires multiple training processes to obtain the best results and this aspect can be highly time consuming.

CONCLUSION

This project showcases the strengths and weaknesses of the convolution neural networks in the field of computer vision under image recognition and classification. The model created will showcase strong results and will work as intended for the image inputs it receives to analyze upon. The future of this project lies in the improvement and advancement of the neural network to make it provide a stronger and more accurate result for a larger dataset as well as improves the time taken to train the network. The biggest challenge for all deep learning models is to solve the time-consuming issue it possesses. So far advancements have not been made into tackling this issue, but there are tremendous amounts of research being aimed in these sectors to solve these issues to make the use of neural networks more viable and easier for anyone who wishes to implement these neural networks into their work.

REFERENCES

1. Md Tohidul Islam, B.M. Nafiz Karim Siddique, Sagidur Rahman and Taskeed Jabid, Image Recognition with Deep Learning, ICIIBMS, Track 2: Artificial Intelligent, Robotics, and Human-Computer Interaction, Bangkok, Thailand, 2018.
2. Fuchao Cheng, Hong Zhang, Wenjie Fan and Barry Harris, Image Recognition Technology based on Deep learning, Springer Science+Business Media, LLC, part of Springer Nature, 2018.
3. Anurag Arnab, Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Mans Larsson, Alexander Kirillov, Bogdan Savchynskyy, Carsten Rother, Fredrik Kahl and Philip Torr, Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation, IEEE Signal processing magazine, January, 2018.
4. Luyu Dong, Fan Chen, Xin Li and Mengting Li, Improved image classification algorithm based on convolutional neural network, Proceedings of 3rd International Conference on E-Business, Information Management and Computer Science (EBIMCS 2020). ACM, Wuhan, China, 4 pages. https://doi.org/10.1145/3453187.3453403, 2020.
5. Xin Jia, Image Recognition method based on deep learning, 2017.
6. Casey Breen, Latifur Khan and Arunkumar Ponnusamy, Image classification using neural networks and ontologies, Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA’02) 1529-4188/02 $17.00 © IEEE, 2002.
7. Meiyin Wu and Li Chen, Image Recognition based on Deep Learning, 978-1-4673-7189-6/15/$31.00©IEEE, 2015.

8. Dwiretno Istiyadi Swasono, Handayani Tjandrasa and Chastine Fathicah, Classification of Tobacco Leaf Pests Using VGG16 Transfer Learning, l2th International Conference on Information & Communication Technology and System (lCTS), 2019.
9. Myeongsuk Pak and Sanghoon Kim, A review of deep learning in Image recognition, This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2015R1D1A1A01057518).
10. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich, Going deeper with Convolutions, 978-1-4673-6964-0/15/$31.00 ©2015 IEEE.
11. Wang Hao, Garbage recognition and classification system based on convolutional neural network VGG16, 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), 2020.
12. Y. Wang et al., Unsupervised local deep feature for image recognition, Information Sciences http://dx.doi.org/10.1016/j.ins.2016.02.044, 2016.
13. Xulei Yang, Zeng Zeng, Sin G. Teo, Li Wang, Vijay Chandrasekha and Steven Hoi, Deep Learning for Practical Image Recognition: Case Study on Kaggle Competitions, KDD ’18, August 19–23, 2018, London, United Kingdom © Association for Computing Machinery. ACM ISBN 978-1-4503-5552-0/18/08. $15.00 https://doi.org/10.1145/3219819.3219907, 2018.