SHIP IMAGE CLASSIFICATION PROJECT DOCUMENTATION

-MAHEK BHARDWAJ

Introduction

Overview

* In this project, I developed a classification algorithm to classify different ship types based on colour (RGB) images of ships. An image of a ship is the input in the classification algorithm, which consists of Convolution neural networks (CNN). Convolutional Neural Networks are a class of deep neural networks, most commonly applied to analyse visual imagery. They are highly useful in terms of classification of image datasets, with high accuracy.

Purpose

* The purpose of the project is to classify different ship images given as input by the user into ten different categories: aircraft carriers, bulkers, cruise ships, fire-fighting vessels, fishing vessels, inland dry cargo vessels, restaurant ships, motor yachts, drilling rigs, and submarines. Our main goal is to achieve at least 80% accuracy with the CNN model and build a User Interface that is easy to use, and gives the result with efficiency. The increased presence of AI systems requires reliable classification algorithms to understand their surrounding environment. These AI systems have the potential to find widespread use in sea and ocean waters, necessitating a reliable classification of their surroundings. Since ships are the most popular means of transportation and warfare in seas and oceans, they need to be classified by autonomous systems.

Literature Survey

Existing problem:

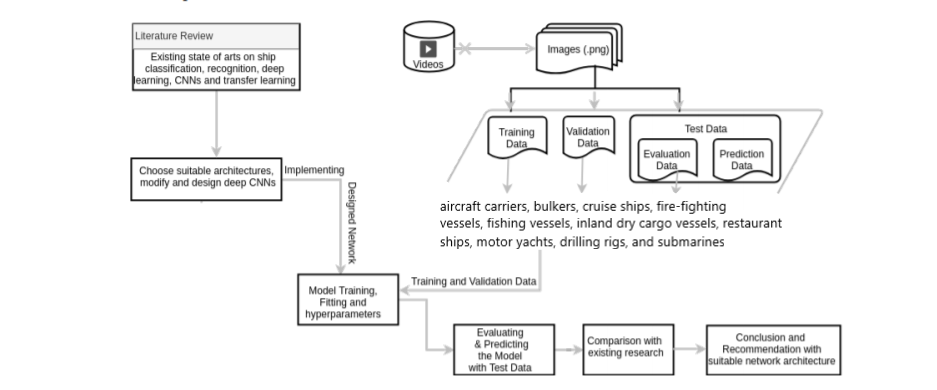
* Oceans are very important for mankind, because they are a very important source of food, they have a very large impact on the global environmental equilibrium, and it is over the oceans that most of the world's commerce is done. Thus, maritime surveillance and monitoring, in particular identifying the ships used, is of great importance to oversee activities like fishing, marine transportation, navigation in general, illegal border encroachment, and search and rescue operations.
* Most of the research on marine ship monitoring involves extraction and classification of image features using different shallow structure algorithms. These techniques are lacking the demanding accuracy and, even if achieved, are at the cost of performances and more workloads in terms of parameters. Advancement in computer vision technology and development of deep learning techniques, particularly convolution neural networks (CNNs) has been offering remarkable performances in the field of image recognition with strong feature learning ability, fewer model training parameters and high recognition accuracy

Proposed solution:

* Advancement in computer vision technology and development of deep learning techniques, particularly convolution neural networks (CNNs) has been offering remarkable performances in the field of image recognition with strong feature learning ability, fewer model training parameters and high recognition accuracy. Thus, using a convolution neural network to classify images of ships into different classes is an effective method to solve the above-mentioned problems.

Theoretical Analysis

 Block Diagram: -



Hardware / Software designing

* **Software used**

1. Jupyter notebooks
2. Spyder
3. Html
4. Anaconda
5. Python programming language

* **Steps:**

1. Data collection
2. Data pre-processing
3. Model building
4. Application building

* **Data collection**

* Training and testing folders were created for model training and testing under one main folder for ship image classification.

* Over 3000 images of ships in the ten different categories were obtained through ShipSpotting.com - 2500 in the training folders and 500 for the testing folders. Each ship category had 250 images in training folders and 50 images in testing folders.

* All images were carefully selected to include nothing but the intended ship in the image. All colour (RBG) images were taken.

* **Data Pre-processing**

* We first import ImageDataGenerator from tensorflow module for data augmentation. This is done for both training and testing data - however, parameters such as zoom\_range, horizontal\_flip, vertical\_flip and width\_shift\_range are only applied to training data. Testing data is only rescaled.

* flow\_from\_directory () is then used to generate batches of augmented data and identify classes automatically from the folder name.

* Data Preprocessing is done.

* **Model Building**

* 4 steps are generally required to build a CNN: Convolution, Max pooling, Flattening, and Full connection:

1. All required libraries are imported from tensorflow.keras.layers and tensorflow.keras.models :  Sequential, Dense, Convolution2D, MaxPooling2D.

1. Convolution: the operation between an input image and a feature detector that results in a feature map is Convolution. We use convolution2D() for creating convolution layers after initializing model using Sequential(). Activation function used was RELU.

1. Max Pooling: Max pooling is to reduce the size of a feature map by sliding a table, and taking the maximum value in the table. A (3,3) filter was used in this model.

1. Flattening: Flattening is to take all pooled feature maps into a single vector as the input for the fully connected layers using model.add(flatten())

1. We now create the hidden and output layers for the model. Units taken are 300 for the first hidden layer and 150 for the rest. Activation function used is Relu, with kernel initializer as “he\_uniform”.

1. For the output layer, units taken are 10, activation function is softmax and kernel initializer is “he\_uniform”.

1. Model is then compiled using the optimizer “adam”.

1. Model is fitted using fit\_generator, with 15 epochs.

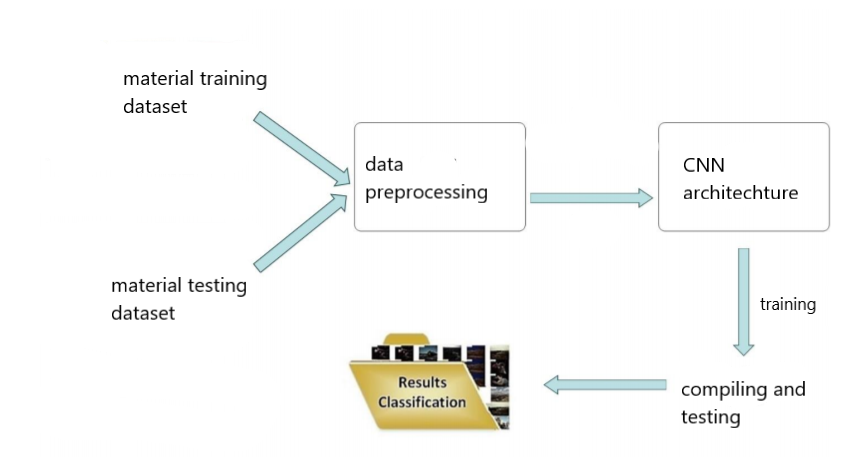
1. Class indices are then generated and a prediction is generated using an image from the testing dataset.

* **Application Building (UI)**

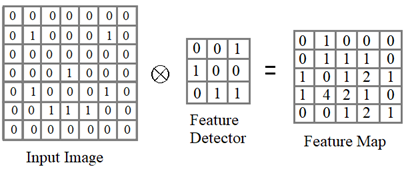
1. A UI was made using a HTML template and flask module.

Flowchart

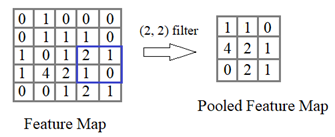
* CNN flowchart:



* Convolution diagram:



* Max Pooling diagram:



Result

* Satisfying results with over 91% accuracy were obtained on the user interface. Images of ships were easily classified into the suitable categories, and the project was successful in achieving its objective.

Advantages & Disadvantages

1. Disadvantages

* When the objects are hidden to a certain extent by other objects or coloured, the human visual system finds signs and other pieces of information to identify what we are seeing. Creating a CNN which has the ability to recognize objects at the same level as humans has been proven difficult. Regardless of where the object is present in the image, a well-trained CNN can identify the object present in the image. But if the object in the image consists of rotations and scaling then the CNN will have a hard time identifying the object in the image.
* A Convolutional neural network is significantly slower due to an operation such as maxpool.

* If the CNN has several layers, then the training process takes a lot of time if the computer doesn’t consist of a good GPU.

* A CNN requires a large Dataset to process and train the neural network.

1. Advantages

* The main advantage of CNN is that it automatically detects important features without any human supervision.

* Little dependence on pre-processing.

* It is easy to understand and fast to implement.

* It has the highest accuracy among all algorithms that predict images.

Applications

* As mentioned in the overview, Ship image classification is highly useful and has several applications. Algorithms to automatically recognize ship type from satellite imagery are desired for numerous maritime applications.

* There is a niche interest in the ability to detect ships in imagery and to recognize their class or type, as part of a broad effort to maintain situational awareness in the world’s oceans.

* Some examples - It may be desirable to localize a 20-foot sailboat on the horizon as captured by a harbour surveillance camera. For another, one may wish to count the number of large merchant ships in a satellite image of a busy port, or to identify each ship down to its hull number. Detecting possible pirate threats as early as possible is also an application.

* This work addresses the question of how well neural network algorithms such as those successful in commercial and academic applications can be applied to ship recognition applications.

Conclusion

* We have successfully created a classification algorithm with CNN using Keras on jupyter notebooks and spyder.

* We have reported results from the application of a convolutional neural network algorithm to ship recognition tasks using images obtained for our dataset.

* We have created a UI which takes an image from the user and classifies it into the following categories: aircraft carriers, bulkers, cruise ships, fire-fighting vessels, fishing vessels, inland dry cargo vessels, restaurant ships, motor yachts, drilling rigs, and submarines

Future Scope

* Ship detection and classification is a widely studied topic both in civilian and military applications. Environmental complexity of marine makes it hard to extract ships from optical images both effectively and efficiently.

* With the increasing volume of satellite image data, automatic ship detection and classification from remote sensing images is a crucial application for both military and civilian fields. However, the detection systems are faced with the need to process massive amounts of incoming data and the requirement of nearly real-time capacity of reaction. Many valuable studies have been carried out in this field, but these typical algorithms are usually effective only for common image analysis. Most of the conventional methods face difficulty in accuracy, performance and complexity.

* Thus, while ship classification using CNN is far ahead of any other classification algorithms, it has great room for improvement in the future, with the idea that the disadvantages mentioned above are solved.

Bibliography

Sources:

* <http://cs229.stanford.edu/proj2017/final-reports/5244159.pdf>

* <https://towardsdatascience.com/image-pre-processing-c1aec0be3edf>

* <https://iopscience.iop.org/article/10.1088/1755-1315/540/1/012049/meta>

* <https://run.unl.pt/bitstream/10362/63805/1/TGEO0210.pdf>

* <https://towardsdatascience.com/cnn-classification-a-cat-or-a-dog-568e6a135602>

* <https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/#:~:text=Keras%20ImageDataGenerator%20is%20a%20gem,up%20on%20the%20overhead%20memory>

* <https://www.mygreatlearning.com/blog/understanding-data-augmentation/>

* <https://arxiv.org/ftp/arxiv/papers/1710/1710.06854.pdf>
* <https://www.researchgate.net/publication/303031905_Convolution_neural_networks_for_ship_type_recognition>

Appendix

 Source code - given along with this word doc in H5 and ipynb file in git repository.

UI output Screenshot:

