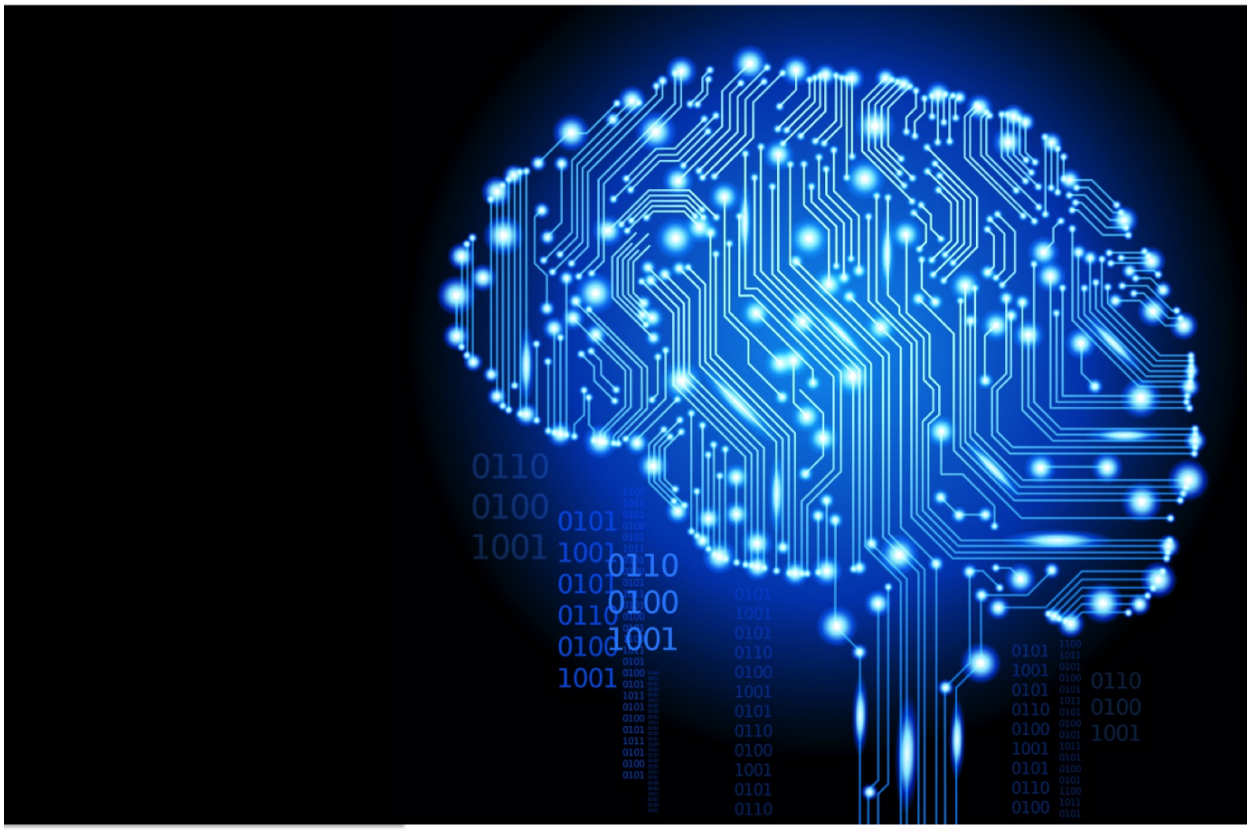

Satellite Image Classification Using Deep Learning

Requirements & Analysis Document

EgSA
وكالة الفضاء المصرية
Egyptian Space Agency

T7COMP

Egyptian Space Agency

Arab Academy for Science, Technology & Maritime Transport

Supervised By Eng. Ahmed ElBaz

Mohamed Ramadan Mohamed

Tarek Abo El-Hagag Hassan

Mostafa Abdel-Karim Mahmoud

Fatma El-Zahraa Ahmed Ali

Mohamed Yasser Omar

ESP
Educational Satellite Program

1 Introduction

In this section we'll discuss the main aspects of the "Satellite Image Classification" System

1.1 Purpose of the system

The purpose of the project is to build a Deep Learning system that takes a number of satellite images and classify each one to one of three categories: [Desert, Plant, Water] and display the result of the classification in a user-friendly interface.

1.2 Scope of the system

Hence it's a pilot project for EgSA training program. The scope of the system is limited to the use of team members and supervisors from EgSA training program.

1.3 Objectives

- Build a CNN model to classify given images with a reasonable accuracy.
- Train the model and save the trained model to be used in prediction
- Build a user-friendly graphical interface
- Connect the trained model with graphical user interface (GUI) so that other people can interact with the model and predict new images

1.4 Definitions, Abbreviations & Conventions

- Model : A schematic description of a system that accounts for its known or inferred properties
- Deep Learning (DL) : is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural
- Convolutional Neural Network (CNN) : In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery.
- Residual neural network (ResNet) : A residual neural network is an artificial neural network of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers.
- Data Augmentation : techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. It helps reduce overfitting when training a machine learning.

- Overfitting: Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

1.5 References

References for building the system will be the following

- EgSA Training Pilot of Project's Statement
- Scene Classification by Ryan Ahmed
- Deep Residual Learning for Image Recognition, Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015

1.6 Overview & Background

Image classification is the process of labeling images according to predefined categories. The process of image classification is based on supervised learning. An image classification model is fed a set of images within a specific category. Based on this set, the algorithm learns which class the test images belong to, and can then predict the correct class of future image inputs, and can even measure how accurate the predictions are.This process introduces multiple challenges, including scale variation, viewpoint variation, intra-class variation, image deformation, image occlusion, illumination conditions and background clutter.

The CNN approach is based on the idea that the model function properly based on a local understanding of the image. It uses fewer parameters compared to a fully connected network by reusing the same parameter numerous times. While a fully connected network generates weights from each pixel on the image, a convolutional neural network generates just enough weights to scan a small area of the image at any given time.This approach is beneficial for the training process—the fewer parameters within the network, the better it performs. Additionally, since the model requires less amount of data, it is also able to train faster.

Residual Neural Network (ResNet) achieved a top-5 error rate of 3.57% and was the first to beat human-level performance on the ILSVRC dataset.ResNet can have up to 152 layers. It uses "skip connections" (also known as gated units) to jump over certain layers in the process and introduces heavy batch normalization. The smart implementation of the architecture of ResNet allows it to have about 6 times more layers than GoogleNet with less complexity.

2 Proposed System

This section will provide a brief overview of the satellite image classification system and then discuss in detail the functional requirements of the system and the nonfunctional ones.

2.1 Overview

The satellite image classification system is a software system developed using machine learning technologies and specifically deep learning and convolutional neural networks to classify images taken by satellites into three categories, desert, plant and water.

The classification model then is implemented in a web application to make it easier for users to interact with the system.

2.2 Functional Requirements

This section describes the high-level functionality of the system and basic system behaviour.

2.2.1 FR1 - Trained Model

The system should train a ResNet18 CNN model on a labeled dataset and save the trained model to be used in prediction.

Data Augmentation should be performed on the dataset before training the model to increase the data and to decrease chances of overfitting.

2.2.2 FR2 - Prediction Server

The system should have a prediction server that uses the trained model to predict and classify the images provided by the system user.

2.2.3 FR3 - The Image Preprocessing

The system prediction server should have the functionality to process whatever image the user sends and adjust it in a way that the prediction model can accept it.

2.2.4 FR4 - Graphical User Interface

The system should have a graphical user interface for the user to interact with and use it’s component to upload the images wanted to be classified.

The user interface can be a web application developed with frontend technologies.

2.2.5 FR5 - Communication

The system graphical user interface should be able to communicate with a prediction server and send the user images to that server and the prediction server should be able to receive these images and perform its mission and send back the results to the GUI.

2.3 Nonfunctional Requirements

This section provides a brief discussion about the nonfunctional system requirements

2.3.1 Usability

- The system should provide a user-friendly interface with no complications.
- The system is organized in such a way that user errors are minimized.
- The system should display the images provided by the user with their respective categories classified by the model in a clear and easy to understand way
- The time needed by the user to classify the images is suitable

2.3.2 Documentation

The developed system should be well documented and that can be done in 2 phases

- The first phase of the project “Pilot of Project” we can rely on the RAD document and the documentation in the source code itself. So the source code should be clean and well commented for anyone to understand quickly and start implementing new features in no time.
- The second phase of the project “Universat” the project should have all the proper documentations, like specification, architecture, design, etc...

2.3.3 Hardware Consideration

The system should be deployable on Clouds, and Hosting services so the code should be optimized and efficient to run the usual services provided online with no need for costly hardware.

A suggested online service to run the prediction model is Google Colaboratory with the following specs :

- 2vCPU @ 2.2GHz
- 13GB RAM
- 100GB Free Space

And an online service to run the prediction model is Heroku with the following specs :

- 1vCPU
- 512MB RAM
- 500MB Free Space

2.3.4 Performance and Efficiency Characteristics

The system should be able to classify images with a reasonable accuracy.
Also both sides of the system should have acceptable performance :

- The web application should load in a reasonable time
- The prediction server should deliver results in a reasonable time, say 1 second maximum for each request of 10 images.

2.3.5 Error Handling

The system should be able to handle error cases like :

- Wrong image format
- The input file isn't an image
- Number of images is too large to be sent to the server
- Image isn't any of the three classes

All of these are scenarios of errors that can happen, so the system should handle each one in a proper way.

2.3.6 System Modifications

The system modifications and updates should be applicable without taking the whole system down, so the system should have a way of continuous integration and continuous deployment.

2.3.7 Privacy Issues

- The system shouldn't store any user relevant information. Even the images being classified shouldn't be stored in any way.
- The system shouldn't violate the users privacy by any means.
- If any files to be stored on the users devices the user should have the option to permit this act or not.

2.3.8 Security Issues

- The system shouldn't expose the users devices to any kind of vulnerability.
- The system should isolate all users using the system to prevent any kind of malicious behaviour

2.3.9 Concurrency Considerations

The system should be able to conduct multiple operations from different users at the same time.

There could be multiple users sending images to be predicted so the system should be able to handle these requests efficiently.

- Average number of users using the system at the same time can be 10 users.
- Each user can send up to 10 images in one request.