Group 2 - Topic Proposal

Automated ship classification from satellite images using Deep Learning

Subject: Deep Learning

Professor: Amir Jafari

Team Members: Akhil Bharadwaj, Shumel Siraj Khan

Overview

The problem selected for this project is the classification of maritime scenes using optical aerial images from the visible spectrum. The goal is to detect and identify various objects in the images, such as ships, land, coast, and sea, to help with maritime monitoring and surveillance. This problem was chosen due to its importance in detecting and preventing criminal activities, accidents, and military attacks in international waters.

About dataset

Link to Download the dataset: Click here

Composition: The MASATI dataset is used for this project, consisting of 6,212 satellite images, which are categorized into seven classes: land, coast, sea, ship, multi, coast-ship, and detail.

Image Details: The images are captured in dynamic marine environments under varying weather and illumination conditions, acquired from Bing Maps in RGB format. The size of the images depends on the region of interest, with an average spatial resolution of around 512 x 512 pixels. The images are stored as PNG files, where pixel values represent RGB colours. The distance between targets and the acquisition satellite varies to obtain captures at different altitudes.

Dataset Organization: To label the category of each image, the dataset is organized into folders, where each folder represents a category. This organization facilitates the efficient handling of images for training and validation purposes.

Adequacy for Training: The dataset is large enough to train a deep network, providing sufficient variation and complexity to build a robust model for maritime scene classification.

Deep Learning Techniques

For this project, various deep learning architectures will be explored. A baseline Convolutional Neural Network (CNN) model will be implemented for image classification. Additionally, pre-trained models like VGG, Inception, and ResNet will be employed to take advantage of transfer learning. A unique architecture that combines a CNN for feature extraction (neural code) and a k-Nearest Neighbour (KNN) model for classification will also be used, based on the work of Antonio-Javier, Antonio Pertusa, and Pablo Gil. The networks will be implemented using the TensorFlow and Sklearn frameworks. Customization may be required to adapt the pre-trained models to the specific classification task and to incorporate the CNN-KNN architecture.

The framework used for implementing the network is TensorFlow with Keras API. This choice is based on the code provided which uses the TensorFlow library for implementing the neural network models like VGG16 and custom CNN. TensorFlow is an open-source library with a comprehensive ecosystem of tools, libraries, and community resources that enables researchers and developers to build machine learning applications easily. Keras, as a high-level API, provides a simpler interface for creating deep learning models, making the development process more user-friendly.

Reference Material to obtain sufficient background and knowledge.

To gain a sufficient understanding of the chosen networks and their application to the problem at hand, the following resources can be used:

- **Official TensorFlow and Keras documentation:** This will provide a solid foundation for understanding the various layers and components of the neural networks, as well as how to implement them using the Keras API.
- **Research papers and articles** on CNNs, VGG16, and other relevant architectures: These will provide valuable insights into the algorithms and techniques used in the neural network models and how they are applied to various problems.

Metric

The performance of the network will be judged based on the following metrics:

- **Accuracy:** This metric is used to determine the percentage of correctly classified images by the model.
- **F1 Score:** This is the harmonic mean of precision and recall and provides a more balanced metric for classification tasks, especially when the classes are imbalanced.
- **Recall:** This measures the ability of the model to identify all the relevant cases within a dataset.
- **Precision:** This measures the proportion of correctly identified positive cases from all the predicted positive cases.

The code provided evaluates these metrics on the train, validation, and test sets to assess the performance of the models.

Rough Schedule

The rough schedule for completing the project is as follows:

April 10-12th: Review the provided code, read the necessary reference materials, and gain a thorough understanding of the chosen networks and their application to the problem. And prepare the dataset, pre-process the images, and split the data into train, validation, and test sets.

April 13-15th: Implement the CNN model and any other additional models (e.g., VGG16) using TensorFlow and Keras. Train the models using the prepared dataset.

April 16th-23rd: Evaluate the performance of the models using the defined metrics. Optimize the models and tune the hyperparameters as needed to achieve better results. And analyse the results, compare the performance of the different models, and draw conclusions. Document the project findings and prepare the final report or presentation.

April 24th: Documentation and GitHub

References

[1] Feng, Y., Diao, W., Sun, X., Yan, M., & Gao, X. (2019, August 14). *Towards automated ship detection and category recognition from high-resolution aerial images*. MDPI. Retrieved April 20, 2023, from https://www.mdpi.com/2072-4292/11/16/1901

[2] J. Alghazo, A. Bashar, G. Latif and M. Zikria, "Maritime Ship Detection using Convolutional Neural Networks from Satellite Images," 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2021, pp. 432-437, doi: 10.1109/CSNT51715.2021.9509628.