Good morning everyone, I am Shan Jacob and I was a student of the course MSc Artificial intelligence in the university of East London. Today I am here to present to you the paper we worked on which is **ANALYSIS OF DEEP NEURAL NETWORKS FOR MILITARY TARGET CLASSIFICATION USING SYNTHETIC APERTURE RADAR IMAGES**. The presenter will be myself, Shan Jacob and the authors of this paper are myself and my professors Dr Julie Wall and Dr Saeed Sharif whom I would like to thank for this oppurtunity and their guidance.

Moving on to the introduction, target detection and classification is very crucial in modern warfare. Thus target classification plays a huge role in military operations.

It is a process that is used by the military for detecting, tracking and identifying enemy targets. This is done through using various types of sensor datas and many analytical methods.

The main advantages this process provides is that it alerts the military personel and enables them to plan military operations while in battle.

Synthetic Aperture Radar systems are those sytems that helps us to create images of high quality or high resolution of the earths surface. This is done by utilizing microwave radiation.

This is done by transmitting a signal to the surface of the earth and later receiving the signal that is reflected back and further processing these signals to create images of the earths surface.

The important advantage of SAR imagery is that they can capture images of the surface in all kinds of scenarios. They can capture high resolution images in all kinds of weather conditions, in all kinds of lightings and even in areas that are covered with forests or clouds or hills.

This phenomenon enables the military to capture images of the enemy targets in these scenarios.

But there are a few challenges in analyzing these images using the traditional methods which includes using human beings to analyze these images or using other machines. This is due to the unique properties that these images posses which are properties like noise and large volumes of data.

This is where Deep Learning is helpful. Deep Learning is a subfield in Machine Learning. Deep Learning makes use of Neural Networks and it learns specific patterns and relationships in a data to analyze or classify these data.

Using Deep Learning for SAR images can help to classify these images more accurately and in a more fast manner.

Using Deep Learning for this process can also automate this process and also help in dealing with large volumes of data.

Thus training Deep learning models on large volumes of SAR images of military targets can help in classifying various types of enemy targets more accurately.

Objectives

The main objective of our paper was to help the military to classify enemy targets by using Deep Learning. Our objective was to use Synthertic Aperture Radar images for this purpose as it can aid the military in detecting and classifying their targets that are situated in all kinds of terrains, any weather condition and lightings. Whether it be day or night, whether they'd be in a dense forest, under cloud cover or even montain shade this technology will aid them in classifying their targets. Here we are classifying 8 different military targets which are 8 different military targets of Russia from their SAR images.

For this, the objective is to use a basic CNN model for the classification and then perform transfer learning using 5 pre-trained models for classification.

Another objective was to use 3 different evaluation metrics which are confusion matrix, classification report and mean average precision for comparing the performance displayed by each of the 6 different models to discover the optimal model for this task.

Liter survey

Now we will look at the related works that other researchers have done in this area

Ryan of Lockheed Martin Space in Pennsylvania also performed classification on the MSTAR dataset. He used the ResNet-18 model for the classification. They also used the confusion matrix for evaluation and achieved a 99% accuracy on the 10 different classes in the dataset. He also further investigated classifying emerging targets using the classifier.

Yan Ouyang and colleagues from Army Engineering University in Nanjing implemented a system for detecting vehicles in the military. So the system that they had developed was based on hierarchical feature representation and reinforcement learning refinement localization. For this purpose, they had constructed a new dataset which they gathered from the internet. The test set in their dataset, they seperated into 3 which are small scale, large scale and the whole dataset. While evaluated using the mean average precision, their system achieved accuracies of 85% for the large set, 66% for the small set and 81% for the whole dataset.

Anishi Gupta and Uma Gupta from Hyderabad implemented a system for real-time-military surveilence. A YOLO model with 58 layers was used for this work. They created a dataset of 20 classes of Pascal VOC and 2 other target classes. They achieved a mean average precision of 79% and 78% for the 2 datasets.

Yuehan Gu and his colleagues from the Shanghai Institute of Satellite have created a system that classifies 3 categories of military vehicles in the MSTAR dataset which are the BTR70, T72 and BMP2. They used MATLAB as programming language for their work and for evaluation they used the confusion matrix. Through their work they classified the targets with a 90% accuracy for the BMP2 and T72 and 70% accuracy for the BTR70.

Ryan of Lockheed Martin Space in Pennsylvania also performed classification on the MSTAR dataset. He used the ResNet-18 model for the classification. They also used the confusion matrix for evaluation and achieved a 99% accuracy on the 10 different classes in the dataset. He also further investigated classifying emerging targets using the classifier and the effect of limited data for training. He experimented different training strategies for this task which include using a scratch model and freezing layers in the model. Further he found out that it is not possible for defining a best method for this task as the availitbility of data is less.

Thus the insights we can get from the related works is that most of the researchers have used normal images for the classification purpose. Only few have investigated using SAR images for the task. Another fact that we can observe is that almost all of the researchers have implemented their research using just one model. While an experimentation of several models for this task was not implemented.

Thus we intend to implement a system that addresses these limitaions. We experiment a different approach by using SAR images for the task and implementing transfer learning and experimenting the performances of diffferent models for the task.

Dataset

So the dataset that is used for this dissertation is the MSTAR dataset or the Moving and Stationary Target Acquisition and Recognition dataset. It was originally created by the US Defense Advanced Research Projects Agency and contains a collection of SAR images of 10 different military targets.

But the dataset that was used here is the MSTAR dataset that is available in Kaggle. This dataset contains images of 8 different military vehicles. Which are vehicles like bulldozers, tanks and trucks.

Now we will look into the deep learning models used for the classification.

The models that were used for classification in this dissertation are all Convolutional Neural Networks. The Convolutional Neural Networks or CNNs are usually used for tasks that involves analysis of videos and images. These CNNs are created by being inspired from the visual cortex that are in animals.

The CNNs they contain several layers. These layers include the convolutional layer, the pooling layer and the fully connected layer.

In the convolutional layer, the model applies filters to the data that is input which are images. By doing this, the model learns the features and patterns that are present in these images.

The pooling layer on the other hand receives the output from the convolutional layer and downsaples the output or it reduces the resolution of the output while preserving the important informations. This is done for reducing the spatial dimenson of the data.

And finally the fully connected layer is the layer that enables the model to make predictions on the basis of the features that were learned. This layer connect all the neurons in one layer to all the neurons in the next layer.

The CNNs are usually used for tasks like object detection, classification of images and other computer vision tasks and they are being used in a very high manner in the fields including facial recognition, medical imaging and others.

VGG16 is a deep convolutional neural network architecture named after the Visual Geometry Group at the University of Oxford, where it was developed. It gained popularity after participating in the

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014, where it achieved excellent performance.

The VGG16 architecture is characterized by its simplicity and uniformity. It consists of 16 weight layers, including 13 convolutional layers and 3 fully connected layers. The convolutional layers use small 3x3 filters with a stride of 1 and a max-pooling layer with a 2x2 window and a stride of 2. The fully connected layers are followed by a softmax activation function for classification.

VGG16 has been widely used as a feature extractor in various computer vision tasks, and its architecture has inspired the development of other deep learning models. However, it is worth noting that due to its depth, VGG16 has a large number of parameters, which can make it computationally expensive and memory-intensive compared to more recent architectures.

The VGG19 architecture is an extension of the VGG16 model. VGG19 comprises a total of 19 layers, which is basically a deeper variant of the original VGG16 model.

InceptionV3 is a convolutional neural network (CNN) architecture developed by researchers at Google. It is part of the Inception family of models and is designed for efficient image recognition and classification. The key innovation of InceptionV3 lies in its use of "inception" modules, which are blocks of layers with different filter sizes and pooling operations. This allows the network to capture information at various spatial scales and learn diverse features simultaneously. It features batch normalization and factorized convolutions, which contribute to faster training and better generalization. InceptionV3, with its innovative inception modules, strikes a balance between accuracy and computational efficiency.

ResNet50, or Residual Network with 50 layers, is a pivotal convolutional neural network architecture developed by Microsoft Research. It stands out for its innovative use of residual connections, addressing the challenges associated with training extremely deep neural networks.

MobileNet is a family of lightweight deep neural network architectures designed for efficient deployment on mobile and edge devices. MobileNet employs depthwise separable convolutions as a key building block. MobileNet architectures are widely used in mobile and edge computing applications, such as image classification, object detection, and facial recognition. They strike a balance between model accuracy and computational efficiency, making them suitable for real-time inference on devices with limited resources.

Confusion matrix

Now we will look at the evaluation metrics that were used in this work. They are 3 which are the confusion matrix, the classification report and the mean average precision.

Here we look at the confusion matrix. It is basically a table that is used to describe a model's performance while classifying.

The matrix is used to compare the predicted labels that the model predicted and the actual labels of the data.

In the confusion matrix the labels predicted by the model is represented in the columns and the actual labels are represented in the rows.

The correct classifiations by a model will be seen in the diagonals and the off-diagonals in the matrix show the misclassified samples.

Classification report

Now looking at the classification report, the classification report displays several parameters of the models performance that are the precision, recall, the F1 score and support.

Precision is basically the proportion of the samples that are actually true positives among what the model classified as positives. And recall can be said as the proportion of true positives among the samples that actually belongs to the class.

Further the classification report also calculated the F1 score which is a harmonic mean of precision and recall and Support that is the total number of samples in each class.

Mean average Precision

Another evaluation metrics that was used here was the Mean average precision. It simply outputs a single number that will summerize the overall accuracy of a model in all the classes.

It is calculated by finding the average of precision values of all classes.

Implementation

Further we move on to the implementation of the work, the dataset downloaded from Kaggle website was split into 80% for training and 20% for testing. The images were rescaled by dividing by 255 for the better performance of the model. This was done using the ImageDataGenerator.

I then used flow_from_directory method to create the training and testing datasets. The size of the input images were set to a tuple of (244,244) and the batch size was set to 32 using the flow_from_directory method.

Further I created a basic CNN model and compiled it using the adam optimizer and trained the model for 20 epochs for a learning rate of 0.001 using the fit method.

Further transfer learning was applied by using the pre-trained models VGG16, VGG19, InceptionV3, ResNet50 and MobileNet which were trained on the imageNet dataset. These were loaded from the Keras deep learning framework.

The top layer of these models which is the fully connected layer was excluded and the pre-trained models weights are freezed.

Further I added a globalaveragepooling layer to reduce the width and height of the feature maps.

These models were also further compiled using the adam optimizer and trained using the fit method for 20 epochs with a learning rate of 0.001.

Results

Now we look into a few examples of the results that were obtained from these models.

Here we see the confusion matrix and classification report obtained from the basic CNN model.

From the confusion matrix, we can observe that the model was able to correctly classify all the samples in 3 classes of the MSTAR dataset which are SLICY, T62, and BRDM_2. The model performed the worst for the classes 2S1 and D7 where it made 145 correct predictions for 2S1 while misclassifying 87 samples and correctly predicted 43 samples of the D7 while misclassifying 71 samples. The model performed reasonably well for all the other classes misclassifying only a few samples for each class.

Looking at the classification report, we can see that the model performs well on most of the classes, with high precision, recall, and F1-score. However, for the class "D7", the recall is particularly low (0.38), indicating that the model is not able to correctly identify many of the positive samples for that class. The class "T62", and " ZSU_23_4" also have a lower precision compared to other classes, having 0.76 precision values each indicating that some of the predicted positive cases are actually negative.

These are the results obtained while the VGG16 model was evaluated.

From the confusion matrix obtained, we can observe that the model performed very well in correctly predicting the classes in the MSTAR dataset. The model had made correct predictions for all the samples in the classes BRDM_2, BTR_60, SLICY, and T62. The model misclassified 14 instances in the 2S1 class but displayed great performance on all the other classes.

By observing the classification report we can find that the model achieved an accuracy of 0.99, indicating that it correctly classified 99% of the examples. The macro and weighted average of the precision, recall, and F1-score metrics are both high (0.99 and 0.99), indicating that the model is performing well across all classes.

Moving on to the discussions, the table displays the performance of all the models used. From the table, we can observe that all the models displayed good performance on the task excluding the ResNet50. The ResNet50 model displayed lower accuracy, lower precision and recall values, a low mean average precision and misclassified many samples compared to the other models.

The model that displayed the highest performance was the VGG16 achieving high values in all the evaluation metrics. But comparing the time taken for training and testing it can be observed that the VGG16 is very slow. Thus, comparing the overall performance of the model on all the evaluation metrics and the time taken for training and testing (3 minutes for training and 46 seconds for testing) it can be concluded that the MobileNet model displayed the best performance on the MSTAR dataset that was used.

Conclusion

In conclusion we implemented a system for classifying SAR images of 8 different military vehicles in the MSTAR dataset. Firstly we used a basic CNN and then applied transfer learning by using 5 pretrained models. The performances of these models were compared using 3 evaluation metrics. And further I concluded that the MobileNet model used was the optimal model for the task considering the performance of the model and other factors.

By further labelling the images in the dataset and implementing further improvements, this system can be used in real-time detection and classification of military targets in warzones.