Remote Sensing Satellite Image Classification Using CNN Architectures Enhanced With Attention Mechanisms

Jothika K
Amrita school of Artificial Intelligence,
Amrita Vishwa Vidhyapeetham,
Coimbatore, India
cb.sc.u4aie23133@cb.students.amrita.edu

Vikasini S
Amrita school of Artificial Intelligence,
Amrita Vishwa Vidhyapeetham,
Coimbatore, India
cb.sc.u4aie23159@cb.students.amrita.edu

Namithaa V
Amrita school of Artificial Intelligence,
Amrita Vishwa Vidhyapeetham,
Coimbatore, India
cb.sc.u4aie23150@cb.students.amrita.edu

Abstract*— This study focuses on improving classification of remote sensing images by using advanced convolutional neural network (CNN) architectures combined with attention mechanisms.Three CNN models were combined with attention mechanism to focus on the important features of the input data. The MobileNetV2 is combined with the Convolutional Block Attention module.The ResNet50 and the Densenet-121 is combined with the Squeeze and Excitation Blocks. Impressively, the MobileNetV2 with CBAM , DenseNet-121 and ResNet50 with SEBlock achieved a test accuracy of 95.39%,89.52% and 93.69% respectively. These results show that how CNN architectures could be improved effectively with by integrating attention mechansims.

Keywords*— Remote sensing satellite image classification, Convolutional neural networks, MobileNetV2, DenseNet-121, ResNet50, Attention mechanisms, Convolutional Block Attention Module (CBAM), Squeeze-and-Excitation Block (SEBlock)

I. INTRODUCTION

Classifying remote sensing images is an important challenge for many applications, such as environmental monitoring, urban planning, and disaster management. Because convolutional neural networks (CNNs) can automatically learn and extract hierarchical features from raw pixel values, they have become an effective tool for image categorization. Nevertheless, it is frequently difficult for conventional CNN architectures to capture the intricate patterns seen in high-resolution remote sensing images. This work investigates the addition of attention mechanism to CNN designs, which enable the model to concentrate on the most pertinent portions of the input data, in order to overcome these difficulties. To train and assess our models, we used the Aerial Image Dataset (AID), which includes thirty distinct scene types. In this study, we allow the CNN models to focus on imwe increase feature extraction by focusing on important portions of the input data by integrating the Convolutional Block Attention Module (CBAM) into the MobileNetV2 architecture. Furthermore, we add Squeeze-and-Excitation Blocks (SEBlock) to the DenseNet-121 and ResNet50 architectures, respectively. By highlighting the most informative aspects and suppressing the irrelevant ones, these attention processes seek to improve the models' representational capacity.

II. DATASET

For training our model, we use the Aerial Image Dataset (AID) which has 30 different scene classes, such as airports, commercial areas, industrial regions, residential zones, farmlands, forests, and more. Each class has between 200 and 400 image samples, which sums up to 10,000 images in total. The images are in RGB color format with a resolution of 600x600 pixels, providing more information required for accurate classification. The AID dataset's wide range and high quality make it apt for training and testing deep learning models.

III. BACKGROUND THEORY

1. MobileNetV2 Architecture

MobileNetV2 is an efficient convolutional neural network (CNN) designed for devices with limited resources, like mobile phones and embedded systems. This design

includes several fundamental advances that greatly lower computational costs and memory use while retaining good accuracy. MobileNetV2 is based on depthwise separable convolutions, inverted residuals, and linear bottlenecks.
Depthwise separable convolutions are a key feature of MobileNetV2. They break down the usual convolution operation into two simpler steps: depthwise convolution and pointwise convolution. Depthwise convolution applies a single filter to each input channel, which reduces the computational load. Pointwise convolution (1x1 convolution) combines these channels to give the final output. Because of this distinction, MobileNetV2 can perform similarly to standard convolutions with much fewer computations.

The architecture also includes inverted residuals and linear bottlenecks. Inverted residuals use shortcut connections to connect layers where the input dimension is bigger than the output dimension.Meanwhile, linear bottlenecks condense the feature representation by lowering the number of channels in the intermediate layers, saving essential information that would otherwise be lost in conventional activation functions. This approach improves the accuracy even with reduced model complexity and thus MobileNetV2 becomes a good solution for image classification tasks.

2. Convolutional Block Attention module (CBAM)

The Convolutional Block Attention Module (CBAM) is added to MobileNetV2 to enhance feature extraction by focusing on the key parts of the input data. CBAM works in two stages: channel attention and spatial attention.

The channel attention module is designed to highlight the most important feature channels. It achieves this by applying global average pooling and global max pooling over the spatial dimensions, which creates two different contextual descriptors.These descriptors go through a shared network of dense layers to produce a channel attention map. This map is then multiplied with the input feature map to prioritize the most relevant channels, helping the model better focus on most required features.

Then , the spatial attention module helps to focus on important parts of the feature map. It takes the output from the channel attention and uses average pooling and max pooling along the channel axis to create spatial descriptors. These descriptors are combined and passed through a convolutional layer to make a spatial attention map.

3. DenseNet-121 Architecture

Recently a lot of attention in the deep learning field is being given to the design of DenseNet (Densely Connected Convolutional Networks) due to its unique way of connection between layers. Dense connections are forwarded from each layer to all subsequent layers from the input, allowing for maximum information exchange between the layers, instead

of traditional convolutional networks, where each layer have a specific set of weights and output. This dense connection pattern is useful for those tasks that rely heavily on deep network architecture as it allows better feature propagation, efficient use of parameters and can even alleviate the vanishing gradient problem.

Dense Blocks: DenseNet121 is made up of four dense blocks, each with several thick layers which are inter connected with each other. By creating dense connections where the output of each layer is concatenated with the outputs of all previous layers inside the same block these blocks are intended to enable considerable feature reuse. This structure improves the model's learning efficiency by greatly enhancing feature propagation and resolving the vanishing gradient problem. In order to provide efficient and varied feature extraction, the dense blocks incorporate layers of 1x1 and 3x3 convolutions, batch normalization, and ReLU activations.

Transition Layers: There are three transition layers dotted within the thick blocks. These layers are essential for lowering the feature maps' dimensionality, which regulates the model's complexity and guards against overfitting. In order to efficiently downsample the feature maps and preserve computational efficiency, each transition layer consists of a 1x1 convolution followed by a 2x2 average pooling layer with a stride of 2.

Initial and Final Layers: An initial convolution and pooling layer forms the foundation of DenseNet121's structural design, as seen in the accompanying diagram. In order to extract the first features from the input picture and reduce its spatial dimensions, the network begins with a 7x7 convolutional layer and moves on to a 3x3 max-pooling layer. A fully linked (dense) layer with a softmax activation function for classification follows the global average pooling layer, which averages each feature map to produce a single value. With this design, the DenseNet121 model is guaranteed to be able to handle the complex and varied nature of satellite images.

4. ResNet50 Architecture

Convolutional neural networks (CNNs), one type of deep learning approach, have made tremendous progress in photo classification. One notable variant among these is the ResNet (Residual Networks) design. ResNet-50's innovative architecture, which addresses the vanishing gradient problem in deep networks, is especially notable. The "vanishing gradient problem" refers to the situation where gradients become extraordinarily small during training, particularly in

deep networks. Early layers obstruct efficient learning since they are hard to update their weights. The ResNet architecture incorporates skip connections—also known as shortcut connections—to overcome this. These skip connections enable gradients to flow through the layers of the network without diminishing by letting the network learn identity mappings.

The skip connections in the ResNet-50 design allow for efficient gradient flow, which enhances the training of very deep networks and permits the model to learn more complex features without running into the problem of vanishing gradients. As such, ResNet-50 has excellent performance in applications that call for intricate categorization scenarios and high-resolution images, such as those seen in the AID dataset. Using the powerful ResNet-50 model, this effort aims to accurately categorize satellite pictures into the relevant scene categories. Numerous applications that depend on satellite image processing will advance as a result.

5. Squeeze-and-Excitation Block (SEBlock)

The Squeeze-and-Excitation (SE) block is the attention mechanism used in the ResNet50 model and the DenseNet121 model for satellite image processing. By emphasizing the most informative traits and suppressing the less relevant ones, this technique improves the representational capacity of the model.

In order for the SE block to function, the interdependencies between the convolutional feature channels must be explicitly modeled. Squeeze and Excitation are the two primary methods used to do this.

Squeeze:

The input feature maps undergo Global Average Pooling (GAP). This process efficiently summarizes the global spatial information of each channel by condensing the spatial dimensions (height and width) into a single number for each channel.A channel descriptor that represents the feature replies' worldwide distribution is the end product.

Excitation:

Two densely connected layers pass the channel description through them. A predetermined reduction ratio, such as 16, is applied to the first layer's channel count, and the original number of channels is restored in the second layer.

To scale the relevance of each channel between 0 and 1, the first dense layer utilizes a ReLU activation function, and the second uses a sigmoid activation function.

As a result, a set of weights reflecting each channel's significance is produced.

Recalibration:

The original feature maps are scaled using the weights derived from the Excitation stage. significant features (high weights) are emphasized while less significant features (low weights) are suppressed throughout this recalibration procedure.

The model may dynamically prioritize channels according to their importance to the particular job, which in this case is satellite image categorization, by including SE blocks into the ResNet50 and DenseNet121 architecture. The model performs better and is more accurate in image classification tasks when it is able to concentrate on the most important aspects of satellite images thanks to this attention mechanism.

IV. METHODOLOGY

1. MobileNetV2 with CBAM model

We prepared our dataset by resizing all the images to 224x224 pixels. Then, we applied some data augmentation techniques like rotation, shifting, shearing, and flipping to make the model more robust.We split the dataset into three parts: 80% for training,10% for validation and 10% for testing. For training the CBAM-MobileNetV2 model, we used the Adam optimizer with a learning rate of 0.0001. The training process ran for 30 epochs with a batch size of 32. We also included early stopping and learning rate reduction to avoid overfitting.We evaluated the model using evaluation metrics such as accuracy,precision,recall,F1-score and mean average precision.

2. DenseNet-121 with SEBlock model

We resized each image to 224 by 224 pixels in order to build our dataset. To strengthen the model, data augmentation methods like rotation, shifting, shearing, zooming, and flipping were used. 20% was set aside for validation and the remaining 80% was used for training. Using a learning rate of 0.0001, we trained the SEBlock-DenseNet121 model using the Adam optimizer. The training procedure had a batch size of 32 and ran for 40 epochs. A number of callbacks were included, such as learning rate decrease, early halting based on validation loss, model checkpointing, and a custom callback to end training when validation accuracy hit 100%. The final 50 layers of DenseNet121 were made trainable, while the preceding levels were frozen, in order to fine-tune the model. We evaluated the model using evaluation metrics such as accuracy,precision,recall,F1-score and mean average precision.

3. ResNet50 with SEBlock model

10% of the AID dataset was set aside for testing, while the remaining 80% was divided across training, validation,

and test sets. To make all of the photographs fit the ResNet-50 model's input size, they were all downsized to 224 by 224 pixels. Using ImageDataGenerator, data augmentation methods including rotation, shifting, shearing, and flipping were applied to the training images to improve the resilience of the model. The images' pixel values were standardized to fall between 0 and 1 in order to guarantee numerical stability throughout training. For effective training, the augmented photos were batched into groups of 32 images each, and for probability distribution prediction, categorical labels were encoded into one-hot vectors.

With a batch size of 32, the model was trained for 30 epochs using the supplemented training data. Early stopping was used to end training if the validation accuracy did not increase after a predetermined number of epochs. Metrics like accuracy, precision, recall, F1-score, and mean average precision were used to evaluate the trained model on the test data in order to gauge its performance on unseen samples.

V. RESULTS AND DISCUSSION

The MobileNetV2 with CBAM model gave a high test accuracy of 95.39%, which outperformed the baseline model. Training accuracy reached a highest of 99.02% with a training loss of 0.0343, while validation accuracy and loss were recorded at 93.80% and 0.344, respectively. These results indicate a well-generalized model with minimal overfitting.

The DenseNet121 with SEBlock model gave a test accuracy of 89.52%, which outperformed the baseline model. Training accuracy reached a highest of 87.92% with a training loss of 0.4368, while validation accuracy and loss were recorded at 89.42% and 0.3615, respectively.

The ResNet121 with SEBlock model gave a test accuracy of 93.69%, which outperformed the baseline model. Training accuracy reached a highest of 98.82% with a training loss of 0.0449 , while validation accuracy and loss were recorded at 94% and 0.3831, respectively.

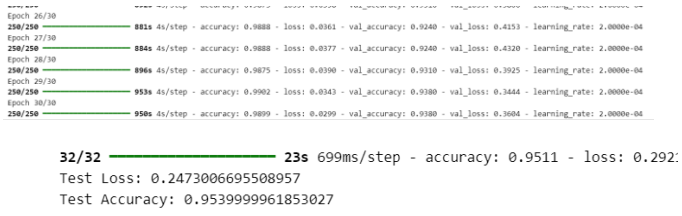

Fig 1. Training the MobileNetV2-CBAM model

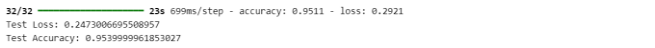

Fig 2. Test accuracy and loss scores of MobileNetV2-CBAM model

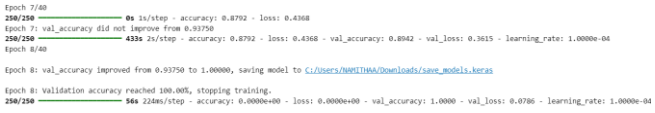

Fig 3. Training the DenseNet-SEBlock model

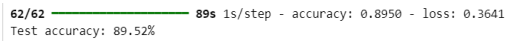

Fig 4. Test accuracy and loss scores of DenseNet-SEBlock model

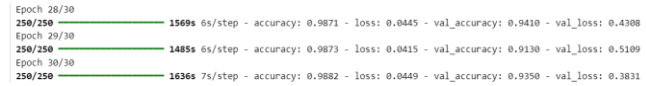

Fig 5. Training the ResNet50-SEBlock model

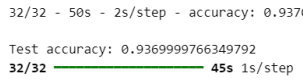

Fig 6. Test accuracy and loss scores of ResNet50-SEBlock model

VI. CONCLUSION

Finally, utilizing the Aerial Image Dataset (AID), this study investigated the efficacy of three widely used convolutional neural network (CNN) architectures for identifying satellite images: ResNet50, MobileNetV2, and DenseNet121. The models were assessed to see if adding attention mechanisms like CBAM and SE Blocks may increase generalization and classification accuracy. The findings suggest that by including these attention mechanisms into CNN architectures, model performance can be improved. To enhance performance and efficiency in remote sensing applications, more research might concentrate on refining these models for particular satellite image categorization tasks.

REFERENCES

[1] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene classification: Benchmark and state of the art,” Proceedings of the IEEE, vol. 105, no. 10, pp. 1865– 1883, 2017.

[2] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, “Deep learning for remote sensing image classification: A survey,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 6, p. e1264, 2018.

[3] Yang, F. An Improved Yolo V3 Algorithm for Remote Sensing Image Target Detection. Journal of Physics: Conference Series 2132(1), 012028 (2021).