Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 8 No. 3 (2024) 389 - 400 e-ISSN: 2580-0760

Remote Sensing Scene Classification using ConvNeXt-Tiny Model with Attention Mechanism and Label Smoothing

Rachmawan Atmaji Perdana[1]*, Aniati Murni Arimurthy[2], Risnandar[3]
[1,2]Computer Science, Faculty of Computer Science, University of Indonesia, Depok, Indonesia
[3]Research Centre of AI and Cyber Security, National Agency for Research and Innovation, Bandung, Indonesia
[1]rachmawan.atmaji21@ui.ac.id, [2]aniati@cs.ui.ac.id, [3]risnandar@brin.go.id

Abstract

Remote Sensing Scene Classification (RSSC) is the discrete categorization of remote sensing images into various classes of scene categories based on their image content. RSSC plays an important role in many fields, such as agriculture, land mapping, and identification of disaster-prone areas. Therefore, a reliable and accurate RSSC algorithm is required to ensure the accuracy of land identification. Many existing studies in recent years have used deep learning methods, especially CNN combined with attention modules to solve this problem. This study focuses on solving the RSSC problem by proposing a deep learning-based method (CNN) with a ConvNeXt-Tiny model integrated with the Efficient Channel Attention Module (ECANet) and label smoothing regularization (LSR). The ConvNeXt-Tiny model shows that a persistent superior outperforms the 'large' model in convinced metrics. ConvNeXt-Tiny model also has a huge advantage in high-precision positioning and higher classification accuracy and localization precision in a variety of complicated scenarios of remote sensing scene recognition. Experiments in this study also aim to prove that the integration of the attention module and LSR in the basic CNN network can improve accuracy because the attention module can strengthen important features and weaken features that are less useful for classification. The experimental results proved that the integration of ECANet and LSR in the ConvNeXt-Tiny base network obtained a higher accuracy of 0.38% in the UC-Merced dataset, 0.7% in the AID, and 0.4% in the WHU-RS19 dataset than the ConvNeXt-Tiny model without ECANet and LSR. The ConvNeXt-Tiny model with ECANet integration and LSR obtained an Accuracy of 99.00±0.41% in the UC-Merced dataset, 95.08±0.20% in AID, and 99.50±0.31% in the WHU-RS19 dataset.

Keywords: *remote sensing scene classification; deep learning; ConvNeXt-Tiny, ECANet; label smoothing regularization*

How to Cite: Rachmawan Atmaji Perdana, Aniati Murni Arimurthy, and Risnandar, "Remote Sensing Scene Classification using ConvNeXt-Tiny Model with Attention Mechanism and Label Smoothing", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 8, no. 3, pp. 389 - 400, Jun. 2024..
DOI: https://doi.org/10.29207/resti.v8i3.5731

1. Introduction

Remote sensing has become a very useful tool in monitoring and analyzing the Earth's environment. With the help of increasingly sophisticated satellite and aerial sensors, high-resolution images that provide an in-depth picture of the Earth's surface can be easily accessed. One important aspect of remote sensing applications is Remote Sensing Scene Classification (RSSC), a complex process that includes automatic identification and categorization of different types of land cover. The utilization of RSSC in the real world is for example for regional mapping, disaster vulnerability detection, environmental and vegetation mapping, and geospatial object detection [1].

The field of RSSC research began in the 1970s with the birth of digital image processing techniques. During this time, remote sensing imagery was acquired with the first Landsat satellites, with low resolution. Processing and analysis of remote sensing imagery at this time was also still limited to the pixel and sub-pixel levels. As the resolution of satellite-generated imagery increased, the research paradigm shifted to analyzing objects in imagery for classification purposes. This change occurred in the early 2000s. Around the beginning of the 2010s, researchers began to realize that the increasing resolution of satellite imagery meant that one image could contain several objects from different classes, making the classification process at the pixel and object level inadequate. This led to research that

Received: 24-02-2024 | Accepted: 18-06-2024 | Published Online: 21-06-2024

attempted to classify remote sensing imagery based on its global context, or in other words, the scene.

Like most pre-deep learning image classification processes, traditional RSSC methods use a combination of handcrafted features and guided machine learning classifiers. Low-level handcrafted features that have been used include Histogram of Oriented Gradient (HoG) [2], Local Binary Pattern (LBP) [3], colour histogram, Scale-Invariant Feature Transform (SIFT) [4], and Gray-Level Co-occurrence Matrix (GLCM) [5]. Some research also employs unsupervised learning methods such as k-means clustering, Principal Component Analysis (PCA), and Autoencoder.

Various state-of-the-art CNN architectures have been widely used in RSSC research such as AlexNet, VGG16, VGG19, GoogLeNet, and ResNet50. CNN models are used as feature extractors and use weight initializations that have been previously trained on ImageNet datasets. Some methods only use transfer learning (e.g. using InceptionV3 [6]), while others perform fine-tuning. Later, Attention Mechanisms were added to the basic CNN network to increase the importance of meaningful features as discriminators for classification. Some of the attention mechanism modules used include CBAM [7], VGG-VD16 with Self-Attention [8], Multiscale Attention Network (MSA-Network) [9], Efficient Channel Attention (ECANet) [10], and Enhanced Attention Network (EAM) [11].

According to [12], there are several problems in RSSC, namely: High semantic variance, in the sense that there are many types/classes of remote sensing images, which are grouped based on the appearance of the earth's surface, such as agricultural areas, highways, airports, settlements, rivers/seas, etc; Low inter-class variance, i.e. the degree of similarity of some classes to others, is quite high (e.g. forest and agricultural areas); High intra-class variance. It was caused by the image acquisition process that varies in scale and angle; Noise caused by differences in atmospheric conditions during the image acquisition process (e.g. clouds).

An example of a case of high intra-class variance is shown in Figure 1. In the top row, we can see the variation in the size of the factory building caused by the difference in image acquisition altitude in the ‘industrial’ class. While in the bottom row, there are various types of soil in the ‘herbaceous crop’ class.

Figure 1. Example of high intra-class variance.

An example of a case of low inter-class variance is shown in Figure 2. ‘Annual crop’ (top row) and ‘permanent crop’ (bottom row) classes have similar rice field structures/patterns.

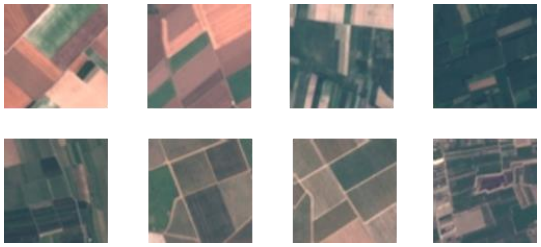

Figure 2. Example of low inter-class variance.

State-of-the-art deep learning architectures continue to evolve. One of the latest deep learning architectures is ConvNeXt [13], which is a modernization of the ResNet architecture. ConvNeXt is developed using the techniques used in Vision Transformer so that its accuracy is improved over the original ResNet model. This research will experiment with the RSSC using the ConvNeXt model integrated with Efficient Attention Network (ECANet) [14] and label smoothing regularization. The accuracy of the RSSC task is projected to improve with the combination of cutting-edge deep learning models, attention mechanisms, and label smoothing.

2. Research Methods

2.1 System Architecture

The ConvNeXt-Tiny model used follows the architecture described in [13]. One ConvNeXt block consists of: A depthwise convolution layer with a kernel size of 7×7 and channels n channels, which match the input feature channels; Layer Normalization (LN) layer; A 1×1 convolution layer with several channels equal to $4n$, activated with GELU; 1×1 convolution layer with the number of channels equal to n. The outcomes of this process will be summed pointwise with the original input features.

The illustration of the ConvNeXt blocks is shown in Figure 3. The overall ConvNeXt-Tiny architecture consists of four stages: the first stage consists of 3 cycles of ConvNeXt blocks, the second stage of 3 cycles, the third stage of 9 cycles, and the fourth stage of 3 cycles. At the beginning of each stage before entering the ConvNeXt blocks, a downsampling process is added which reduces the length and width of the input features to half and doubles the number of channels. This process is illustrated in Figure 4. The effort to “modernize” the standard ResNet model into ConvNeXt is not limited to modification of its architecture, but also improving the training technique by using 300 epoch and AdamW optimizer. Researchers also use a few data augmentation techniques beyond standard techniques such as MixUp, CutMix, and RandAugment. Stochastic Depth and Label Smoothing regularization techniques are also incorporated, resulting in competitive performance on

standard image classification tasks (ImageNet). In this experiment, we will use the ConvNext-Tiny model with pre-trained weights.

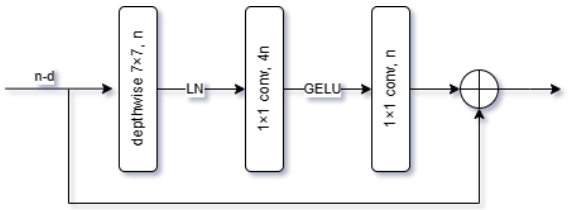

Figure 3. ConvNeXt Block Architecture

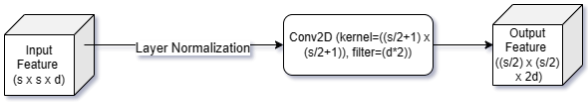

Figure 4. Downsampling process before the ConvNeXt stage

The development of the Attention Mechanism is based on human characteristics/habits in viewing information, especially in the context of images. When viewing information, humans will usually only focus on observing the important parts, such as separating the foreground from the background. Attention Mechanism is very useful for object classification and detection problems, as it can corroborate the features that are important in such contexts by learning the images.

The earliest literature on Channel Attention is the Squeeze-and-Excitation Network (SE-Net) [15]. Although the SE-Net architecture is very simple, adding it to existing CNNs won the ImageNet competition in 2017. At its core, SE-Net consists of two consecutives Fully Connected (FC or Dense) with two types of activation functions, ReLU in the first FC and sigmoid in the second FC. The input for the block FC is Global Average Pooling (GAP) based on the channels of the feature map. Then, the operation result of the block FC is channel-based multiplied over the initial feature map. This is the final result of the SE-Net and can be considered as the importance weighting for each channel.

ECANet is an attention mechanism architecture that is a modification of SE-Net. Researchers argue that one of the weaknesses of SE-Net is that the dimensionality reduction process after Global Average Pooling (GAP), although reducing complexity, removes correspondence information or relationships between channels. In this study, they devise a model which consists of one Fully Connected layer after GAP named SE-Var3, so that it learns the correspondence between one channel and all other channels. This model produced higher Top-1 and Top-5 accuracy than vanilla SE-Net. SE-Var3 was further modified to reduce its complexity by not considering the correspondence of one channel to all other channels, but only to the k neighboring channels. This operation can be easily done using one-dimensional convolution. The results of the 1D convolution are then activated using sigmoid and are multiplied elementwise with the initial feature map χ.

The entire process of the calculation of ECANet can be expressed as in Formula 1.

$$\boldsymbol{F}_{\mathrm{ECA}}(\chi) = \sigma(\mathbf{Conv1D}_k(\mathbf{GAP}(\chi))) \otimes \chi \quad (1)$$

The next question is how to determine the optimal value of k. Here they propose Formula 2, which is formulated based on the assumption that the number of channels C or tensor dimensions input to the GAP is usually a power of two. Meanwhile, the values of the γ and b are obtained through experiments ($\gamma = 2$ and $b = 1$). ECANet produces higher accuracy than SE-Net and CBAM [15] and also has a smaller complexity and number of parameters.

$$k = \psi(C) = \left|\frac{\log_2(C)}{\gamma} + \frac{b}{\gamma}\right|_{\mathrm{odd}} \quad (2)$$

In essence, we embed the ECANet module into the existing backbone CNN, and during the training the weights of the $\mathbf{Conv1D}_k$ layer is learned. The weights $\mathbf{Conv1D}_k$ of represents the importance degree of each neighboring channel. With this, we could direct the backbone network to give more focus to high-importance channels. The diagram of the ECANet Module is shown in Figure 5.

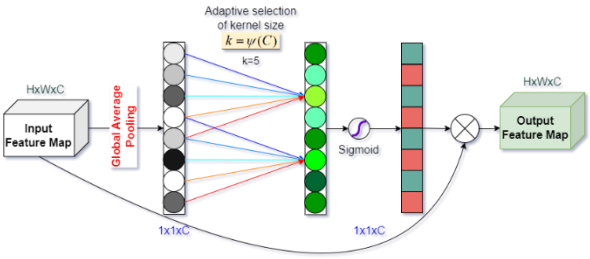

Figure 5. Diagram of ECANet Module

Label Smoothing Regularization (LSR) is a method utilized in machine learning, particularly in classification tasks, to enhance model generalization and mitigate overfitting. In the usual setup, where models predict probabilities for each class, label smoothing diverges by assigning a small probability to incorrect classes during training. This method was first introduced in the paper on Inception architecture [16], and the idea to use LSR for RSSC tasks was inspired by [17].

The main idea behind the LSR is to not make the model too confident because of the one-hot encoding of the input. The 'crisp' encoding like this would ignore every other class (which is not the target class of the sample) during the calculation of the cross-entropy loss function. Meanwhile, in the case of RSSC, a scene image probably consists of multiple objects of different classes, where the class of the most dominant object becomes the assigned label of the image. LSR will re-label the encoding of the input with Formula 3.

$$y_i = \begin{cases} 1 - \varepsilon; i = c \\ \frac{\varepsilon}{k-1}; i \neq c \end{cases} \quad (3)$$

i denotes the class number, y_i is the ground-truth label of class i, c is the target class, k is the total number of

classes, and ε is the smoothing factor. We will use $\varepsilon = 0.1$ in this experiment. This formula allocates a softened probability to both the true class and other classes. The addition of LSR is expected to make the model more robust and have anti-noise ability.

Inspired by the research by [7], in this experiment, ECANet is integrated after each stage of ConvNeXt-Tiny. The integration of ECANet after the ConvNeXt-Tiny stage enhances the features and removes repetitive or redundant information. Research [18] argues that attention to the channel and or spatial aspects after each stage is important because the length and width of the features at the end of the stage are reduced to half of the beginning of the stage, as well as the number of channels being doubled. Weighting each channel and or spatial position of the feature will enhance the feature more effectively so that the next stage of ConvNeXt can learn better input features.

Figure 6 shows the illustration of the ConvNeXt-Tiny architecture integrated with ECANet, where ECANet is inserted after each stage. We will use that architecture in this experiment.

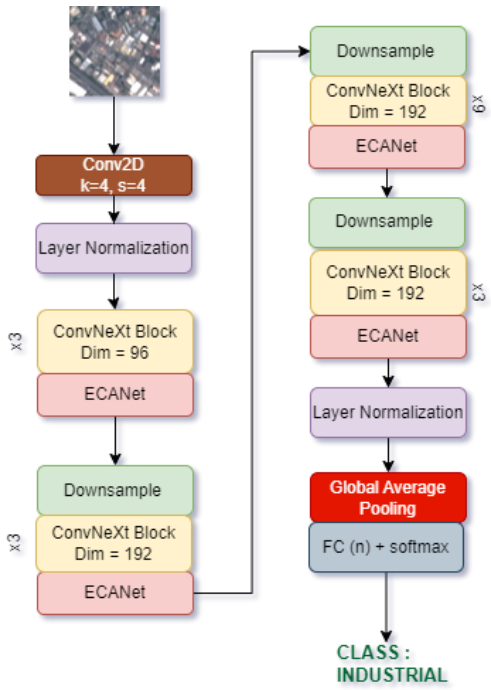

Figure 6. ConvNeXt-Tiny integration with ECANet.

2.2 Dataset

This study uses three public datasets namely UC-Merced [19], Aerial Image Dataset (AID) [20], and WHU-RS19 [21]. The characteristic comparisons of each dataset are shown in Table 1. Augmentation in remote sensing scene classification is important to address one of the main problems of high intra-class variance. The augmentation methods applied in this experiment are rotation, translation, zoom, and random brightness changes. Augmentation by rotation is performed according to the characteristics of the remote sensing image dataset which is taken in various directions and from different heights. Translation is done to obtain a new viewpoint of the image. Zooming is done to obtain variations in image resolution. Meanwhile, changing the brightness level is done to obtain variations in the illumination of a remote sensing image, or it can also be considered as simulating various weather conditions and image acquisition times.

Table 1. Comparison between datasets

Category	UC-Merced	AID	WHU-RS19
Year Released	2010	2016	2012
Number of Class	21	30	19
Number of Images per Class	100	Varied, 200-420. 10000 in total.	Varied, 50-60
Image Size	256×256	600×600	600×600
Pixel Resolution	1 foot	0.5-8 meter	At least 0.5 meter
Acquisition Source	USGS (US images only)	Google Earth (compiled from various sources)	Google Earth (compiled from various sources)

Two augmentations are performed in AID, while in WHU-RS19 three augmentations are performed, and four in UC-Merced. This consideration is based on the number of images in both datasets, where AID has a much larger number of original images. The dataset would be very large and slower to train the model if we performed too many augmentations. UC-Merced has one more augmented image per image than WHU-RS19 because it has a larger number of images per class. Hence, more augmentation is necessary to capture more variances in the dataset. Rotation augmentation performed is in the range of -30° to 30°, translation on the x-axis and y-axis with a range of 5 to 40 pixels, and zoom in the range of 1 to 1.15 times. Brightness augmentation is performed in the range of -30 to 30 levels.

2.3 Experiment Settings and Evaluation Method

Experiments on the three datasets described were conducted using different training-testing ratios according to the ratios widely used in various studies for these datasets. For UC-Merced, a ratio of 80:20 was used [22], AID 50:50 [17], and 80:20 for WHU-RS19 [23]. In this research, the cross-validation technique is used in training the model. The goal is to produce more stable accuracy, avoid overfitting, produce better generalization, and be able to estimate model performance more accurately. The UC Merced and WHU-RS19 datasets use 5-fold cross-validation, while AID uses 2-fold cross-validation. The division of data in k-fold cross-validation uses the Stratified K-Fold method from the Scikit-Learn Library so that the ratio of each class in each fold is the same as the ratio of each class in the overall dataset.

This research will compare the performance of the ConvNeXt-Tiny that is not integrated with ECANet (**Model A**), ConvNeXt-Tiny integrated with ECANet (**Model B**), and ConvNeXt-Tiny integrated with ECANet and using Label Smoothing Regularization

(**Model C**) on the task of RSSC. In the Results and Discussions section, we will only discuss and analyze the results of models A and C. Model B is only used as an ablation study, meaning that we prove that the label smoothing regularization (in Model C) does indeed improve the accuracy of Model B. The performance of each trained model is evaluated using the classification Accuracy and confusion matrix (CM) metrics. The source code is written in Python 3.10.12, and the deep learning framework used is Keras 2.13.1 on top of Tensorflow 2.13.0 runs on NVIDIA Tesla GPU. The hyperparameter settings for each dataset are shown in Table 2.

Table 2 A hyperparameter was used in this experiment.

Hyperparameter	UC-Merced	AID	WHU-RS19
Image size	256×256	256×256	256×256
Batch size	32	24	24
Optimizer	Adam	Adam	Adam
Number of epochs	25	25	50
Loss function	Categorical cross-entropy	Categorical cross-entropy	Categorical cross-entropy

3. Results and Discussions

3.1 Results on UC-Merced

As explained in Section 2, experiments on the UC-Merced dataset used 5-fold cross-validation with 80% training and 20% testing data. The amount of training data for each fold is 8400, and the testing data is 420. The amount of training data is 20 times more than the testing data because the original images in the training set are augmented 4 times. The original images in the testing set are not augmented. Testing accuracy for each fold is shown in

Table 3.

Table 3 Test Set Accuracy of all models on UC-Merced*

Fold Number	Model A	Model B	Model C
1	99.286	99.286	99.524
2	99.048	98.571	99.286
3	98.571	99.524	99.048
4	98.333	98.810	98.810
5	97.857	97.857	98.333
Mean	98.619	98.810	99.000
Standard Deviation	0.509	0.583	0.410

*) X. Wang, "Improving Bag-of-Deep-Visual-Words Model via Combining Deep Features With Feature Difference Vectors," IEEE Access, vol. 10, pp. 35824–35834, 2022, doi: 10.1109/ACCESS.2022.3163256.

Table 3 shows that the ConvNeXt-Tiny model integrated with ECANet (Model B and C) could improve the performance of the ConvNeXt-Tiny model without ECANet (Model A). The addition of label smoothing regularization makes it even more robust. Model A gets 100% accuracy on 10 classes out of 21. However, it has the lowest accuracy for the 'dense residential' class with only 89%. Some images in this class were classified as 'mobile-home park', 'medium residential', 'buildings', and 'storage tanks'. This is due to a similar characteristic between these classes. The 'dense residential', 'medium residential', and 'building' classes all have dominant objects in the form of buildings, which can confuse the classifier. The misclassification to 'storage tanks' occurs because the shape of residential buildings in the image is similar to the warehouse buildings that usually accompany storage tanks. Some of the misclassified images from this model are shown in **Error! Reference source not found.**.

Figure 7 Some misclassified images of Model A in the UC Merced dataset.

Model C can improve the accuracy of the UC-Merced dataset, especially in the 'dense residential' class. The model was able to increase the accuracy of the 'dense residential' class by 8%. In this model, there are no more misclassifications that put the 'dense residential' class image into the 'mobile-home park' and 'storage tanks' classes. Model C gets 100% accuracy on 11 classes out of 21. This model got the lowest accuracy in the 'medium residential' class with only 94%. The problem of high intra-class variance also causes the model to misclassify green-coloured rivers as forests. Some of the misclassified images from this model are shown in Figure 8.

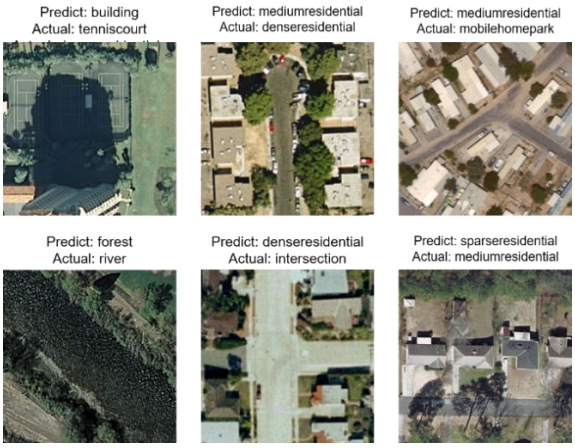

Figure 8 Some misclassified images of Model C in the UC Merced dataset.

Model A produced an average accuracy of 98.62±0.51%. The Confusion Matrix for this model is shown in Figure 9. Model C produced an average

accuracy of 99.00±0.41%. The Confusion Matrix for this model is shown in Figure 10.

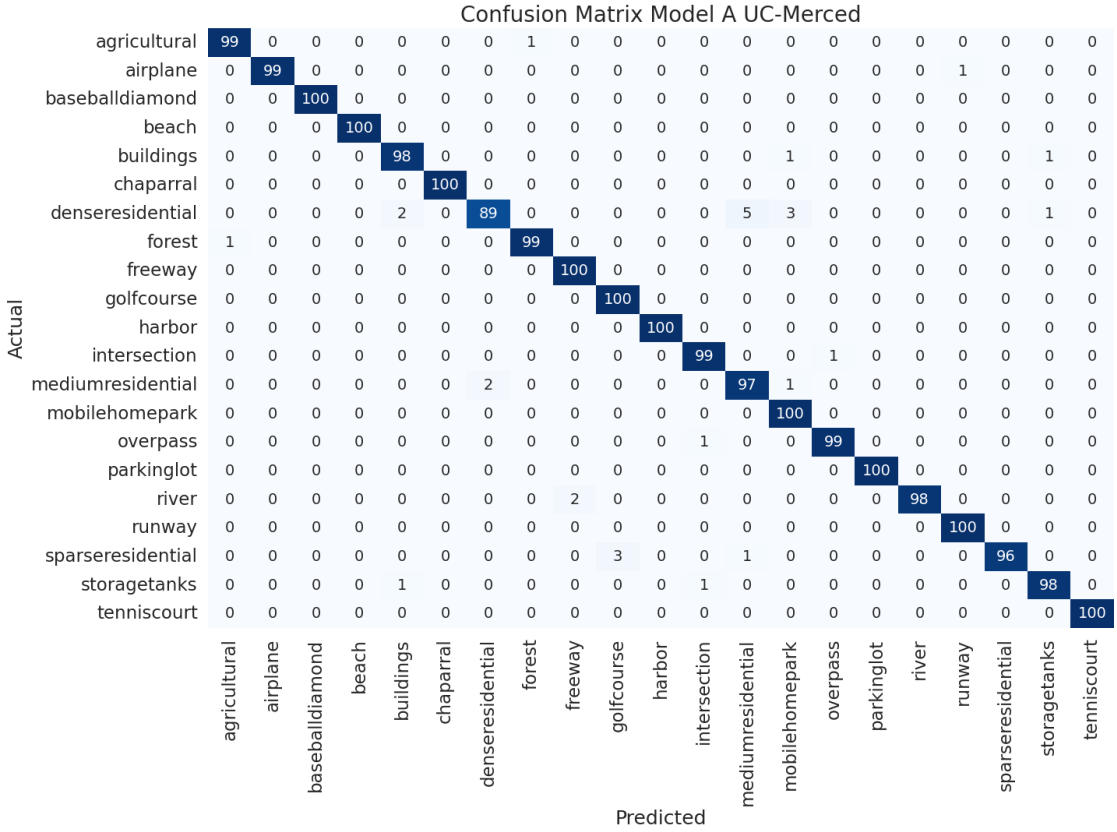

Figure 9 Confusion Matrix of Model A on UC-Merced dataset (in percentages).

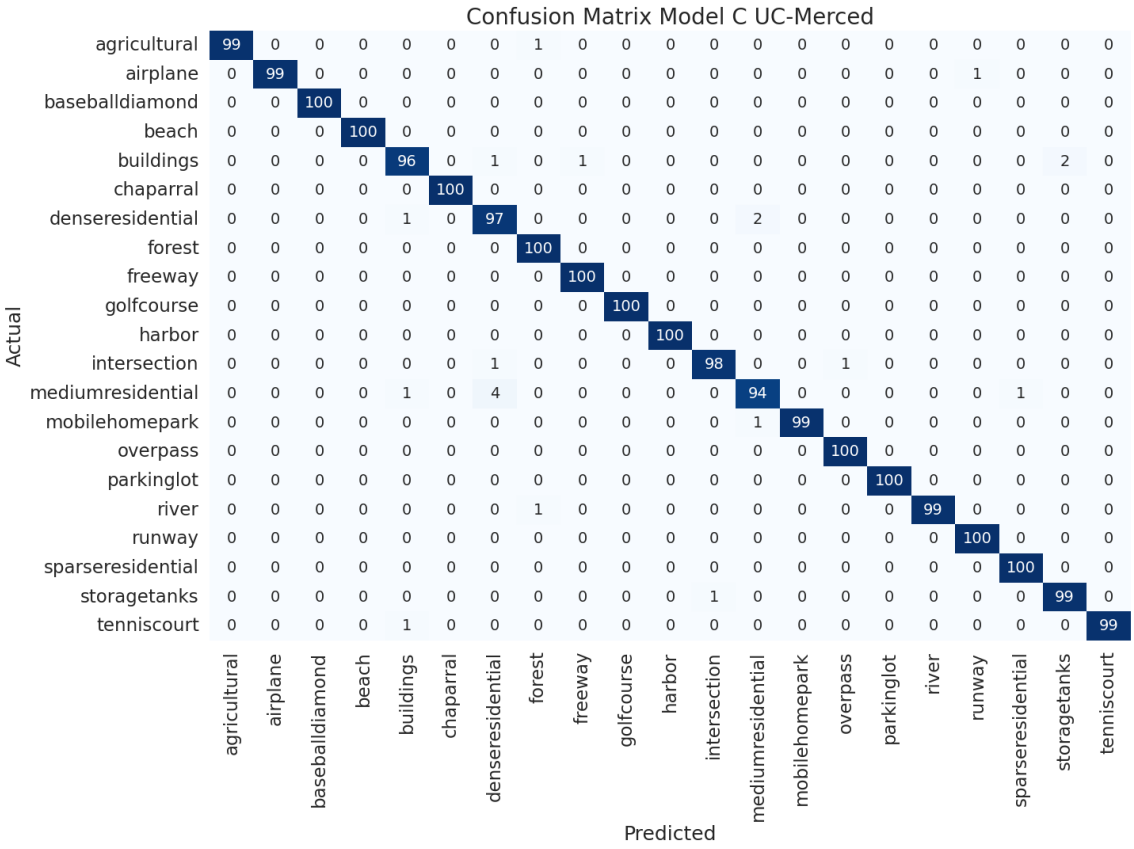

Figure 10 Confusion Matrix of Model C on UC-Merced dataset (in percentages).

3.2 Results on AID

Experiments on the AID dataset used 2-fold cross-validation with 50% training and 50% testing data. The amount of training data for each fold is 15000, and the testing data is 5000. The amount of training data is 3 times that of testing data because the original images in the training set are augmented twice, while the original images in the testing set are not augmented. The testing accuracy values (in per cent) at each fold are shown in Table 4.

Table 4 Test Set Accuracy of all models on AID*

Fold Number	Model A	Model B	Model C
1	94.58	95.06	95.28
2	94.12	94.58	94.88
Mean	94.35	94.82	**95.08**
Standard Deviation	0.23	0.24	0.20

*) G.-S. Xia et al., "AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification," IEEE Transactions on Geoscience and Remote Sensing, vol. 55, pp. 3965–3981, Feb. 2017, doi: 10.1109/TGRS.2017.2685945.

AID has the highest number of classes and images compared to the other two tested datasets, making it the most challenging for classification. The amount of training and testing data used in training on this dataset was also the highest compared to the other two datasets. Despite this, the accuracy is still low, hovering around 94%-95%, while in the other two datasets it has touched 99%. This is mainly due to the low variance between classes, causing the visual appearance of a scene in a particular class to be similar to that of a scene in another class. For example, an image of a large commercial area, when taken from a high altitude, will appear like a collection of small buildings, akin to a dense residential area. Model A produced an average accuracy of 94.35±0.23%. Some of the misclassified images using this model are shown in Figure 11, and the Confusion Matrix for this model is shown in Figure 13.

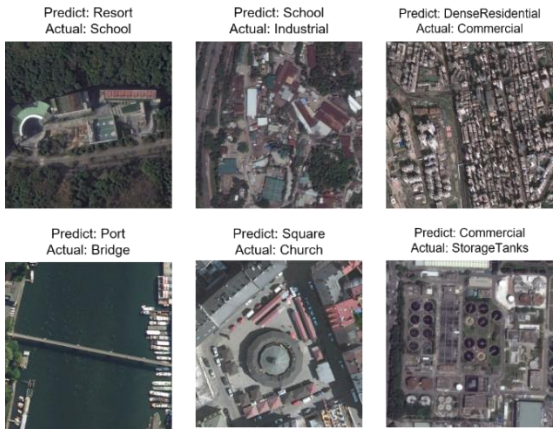

Figure 11 Some misclassified images of Model A in AID

The accuracy for each class in both models did not touch 100%, with the maximum accuracy in Model A being 99.76% in the 'viaduct' class, and the minimum accuracy at 75.33% in the 'school' class. The high intra-class variance of the 'school' class makes its appearance quite similar to 'commercial', 'industrial', and 'church'. In Model C, the maximum accuracy is also achieved in the 'viaduct' class of 99.76% and the minimum is in the 'resort' class of 79.31%. In this model, 10.69% of 'resort' images are predicted as 'park'. The accuracy of Model C is 0.73% higher than Model A, meaning that it could reduce 73 misclassification errors produced in Model A.

The Model C produced an average accuracy of 95.08±0.20%. Some of the misclassified images using this model are shown in Figure 12, and the Confusion Matrix for this model is shown in Figure 14.

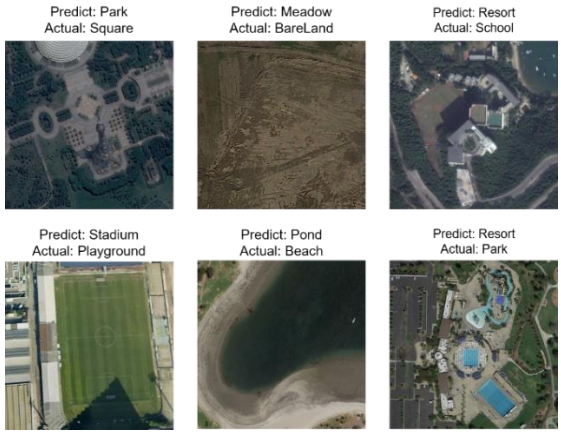

Figure 12 Some misclassified images of Model C in AID

3.3 Results on WHU-RS19

Experiments on the WHU-RS19 dataset used 5-fold cross-validation with 80% training and 20% testing data. The amount of training data for each fold is 3216, and the testing data is 201. The amount of training data is 16 times more than the testing data because the original images in the training set are augmented 3 times. The testing accuracy values (in per cent) at each fold are shown in Table 5.

Table 5 Test Set Accuracy of all models on WHU-RS19*

Fold Number	Model A	Model B	Model C
1	99.005	99.502	99.502
2	99.502	99.005	100.000
3	99.502	99.502	99.502
4	98.010	98.01	99.005
5	99.502	98.507	99.502
Mean	99.104	98.905	**99.502**
Standard Deviation	0.580	0.580	0.315

*) Y. Yang and S. Newsam, "Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification," in ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), 2010.

The WHU-RS19 dataset has its own challenges, namely the small amount of testing data (201), causing the misclassification of just one image to cause a decrease in accuracy by approximately 0.5%. The accuracy of Model B in Fold 5 is one per cent lower than Model A, resulting in a lower mean accuracy. Among the three datasets used in this experiment, this is the only occurrence where Model B obtained lower mean accuracy than Model A. Model C obtained 100% accuracy in fold 2, contributing to the high mean accuracy produced by this model.

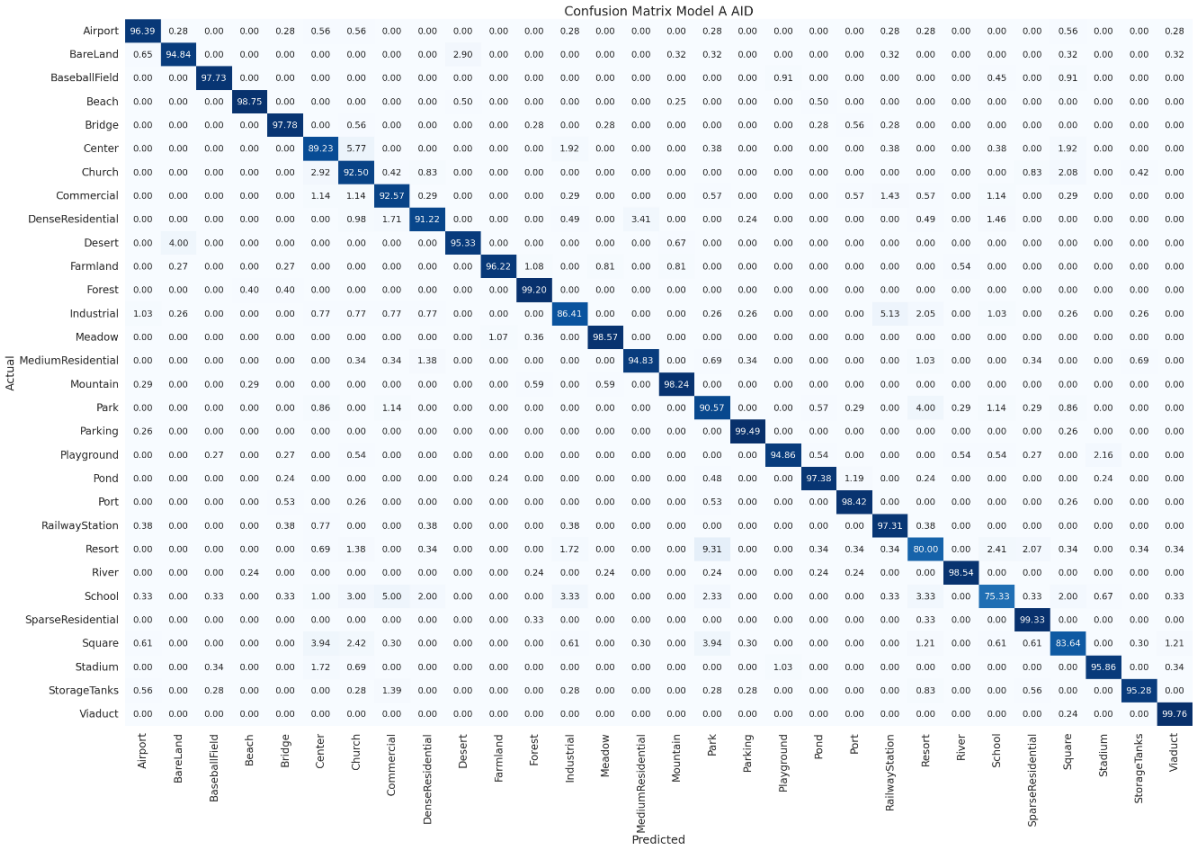

Figure 13 Confusion Matrix of Model A on AID (in percentages)

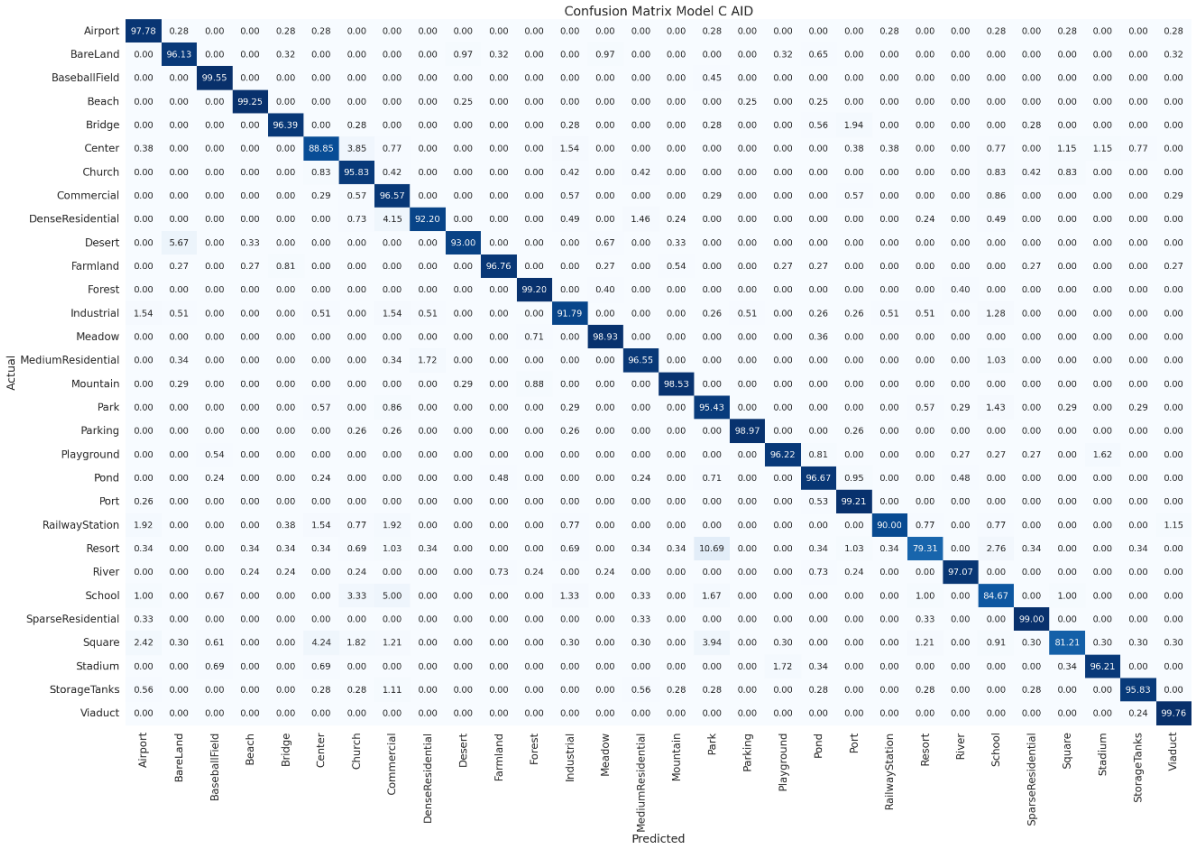

Figure 14 Confusion Matrix of Model C on AID (in percentages)

In Model A, the most misclassifications occurred in the 'Forest' class which was predicted as 'Mountain', 'Port' which was predicted as 'Airport', and 'Residential' which was predicted as 'Commercial'. The characteristics of 'residential' and 'commercial' scenes, which consist of buildings, make some images quite difficult to distinguish. Meanwhile, the misclassification of 'Forest' as 'Mountain' is caused by some images of mountain scenes having dense and green vegetation. Some of the misclassified images using this model are shown in Figure 15.

Figure 15 Some misclassified images of Model A in the WHU-RS19 dataset.

Model A produces an average accuracy of 99.10±0.58%. The Confusion Matrix for this model is shown in Figure 18.

The misclassification of the image of class ‘Forest’ into ‘Mountain’ and ‘Residential’ into ‘Commercial’ is still prevalent in Model C. But overall, the mean accuracy of Model C is higher than Model A. Misclassification of class ‘Beach’ into ‘Desert’ and ‘Airport’ into ‘Port’ does not occur anymore. Model C also reduces the number of misclassifications for the 'Railway Station' class. Some of the misclassified images using this model are shown in Figure 16.

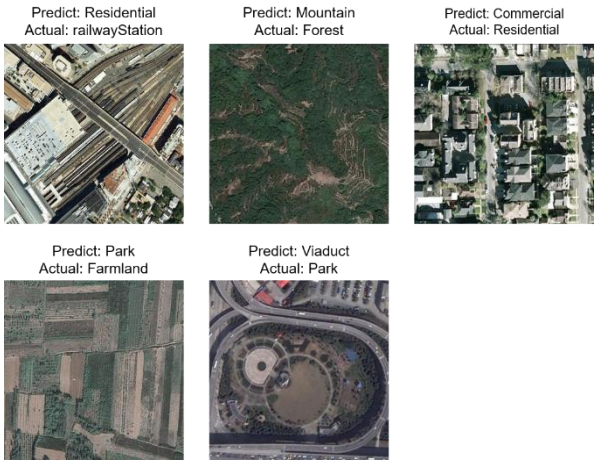

Figure 16 Misclassified images of Model C in the WHU-RS19 dataset.

Model C produced an average accuracy of 99.50±0.31%. The Confusion Matrix for this model is shown in Figure 19.

3.4 Discussions

This experiment shows that the integration of ECANet module after each stage of ConvNeXt-Tiny, equipped with label smoothing regularization (LSR) can improve the ability to learn basic CNN (ConvNeXt-Tiny) features by giving more attention to important features and reducing the importance of unimportant features. This is demonstrated by the Accuracy value obtained by the ECANet-integrated ConvNeXt-Tiny model with LSR (Model C) which is higher by 0.38% on the UC-Merced dataset, 0.73% on the AID dataset, and 0.4% on the WHU-RS19 dataset compared to the ConvNeXt-Tiny model which is not integrated with ECANet (Model A). We also do the ablation study via Model B, which consists of ConvNeXt-Tiny backbone with integration of ECANet but not using LSR. In this experiment, Model B has a slightly lower mean accuracy (0.2%, 0.26%, and 0.6%) than Model C, which proves that the addition of LSR plays a role in improving accuracy. The summary of our experiment is shown in Figure 17.

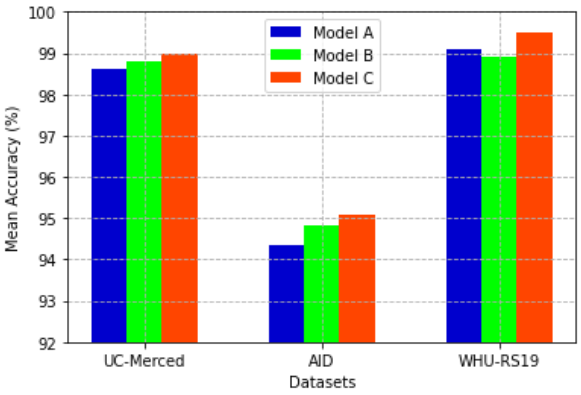

Figure 17 Overall Accuracy of Model A, B, and C on UC-Merced, AID, and WHU-RS19 datasets.

Our proposed model is a new architecture integrating attention mechanism with backbone CNN models, in this case, state-of-the-art models. In this section, we also show how our accuracy compares with other literature for each dataset. The comparisons in UC-Merced, AID, and WHU-RS19 are shown in Table 6,

, and Table 8, respectively.

Table 6 Comparison of Accuracy in UC Merced with Other Literature

Model	Accuracy (%)
Compact Deep Color Features [24]	97.40±0.62
ResNet-50+EAM [11]	98.98±0.37
VGG-VD16+SAFF [8]	97.02±0.78
ResNet-101+CBAM [15]	98.71±0.46
EfficientNetV2L [25]	97.38
VGG-VD16+MICP+multi-size [26]	98.70±0.43
InceptionV3 (Transfer Learning) [6]	98.30
RSCNet [17]	99.05
Model C (ours)	99.00±0.41

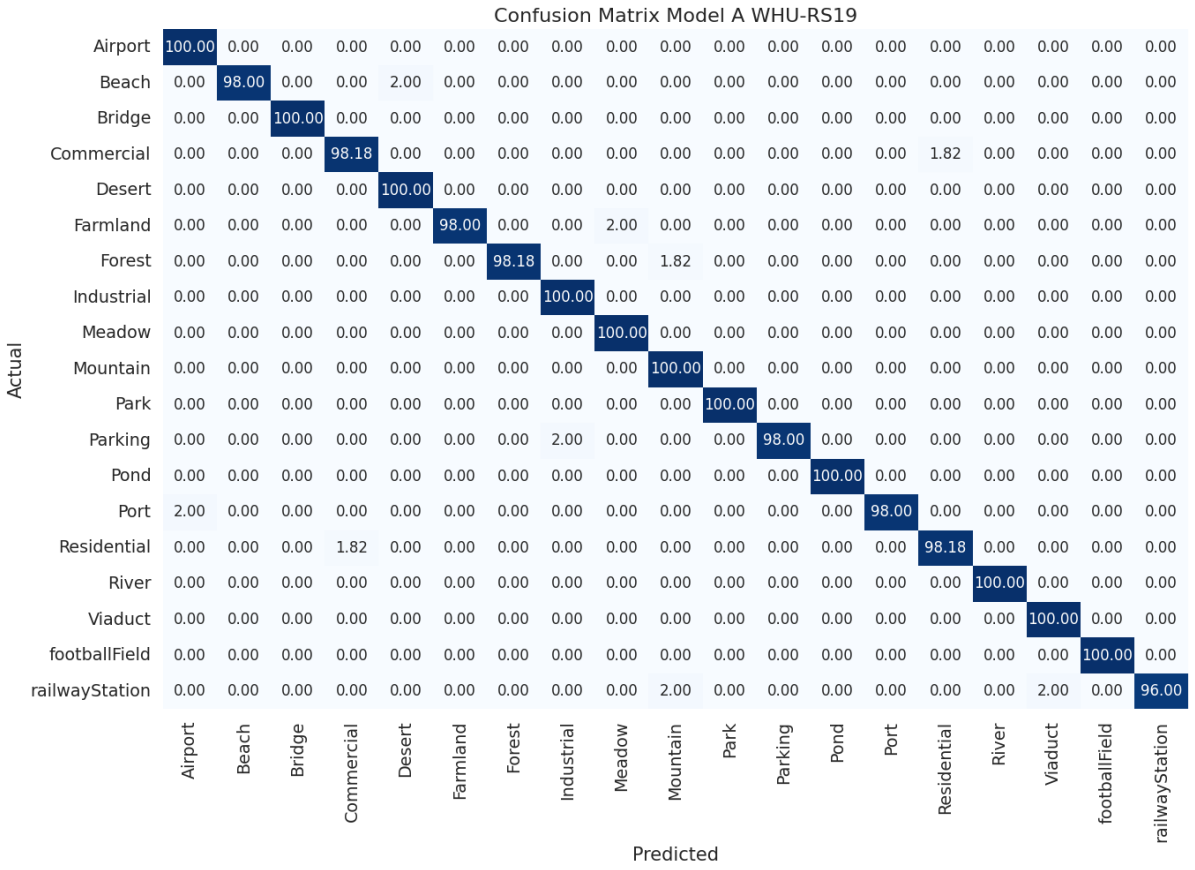

Figure 18 Confusion Matrix of Model A on WHU-RS19 dataset (in percentages)

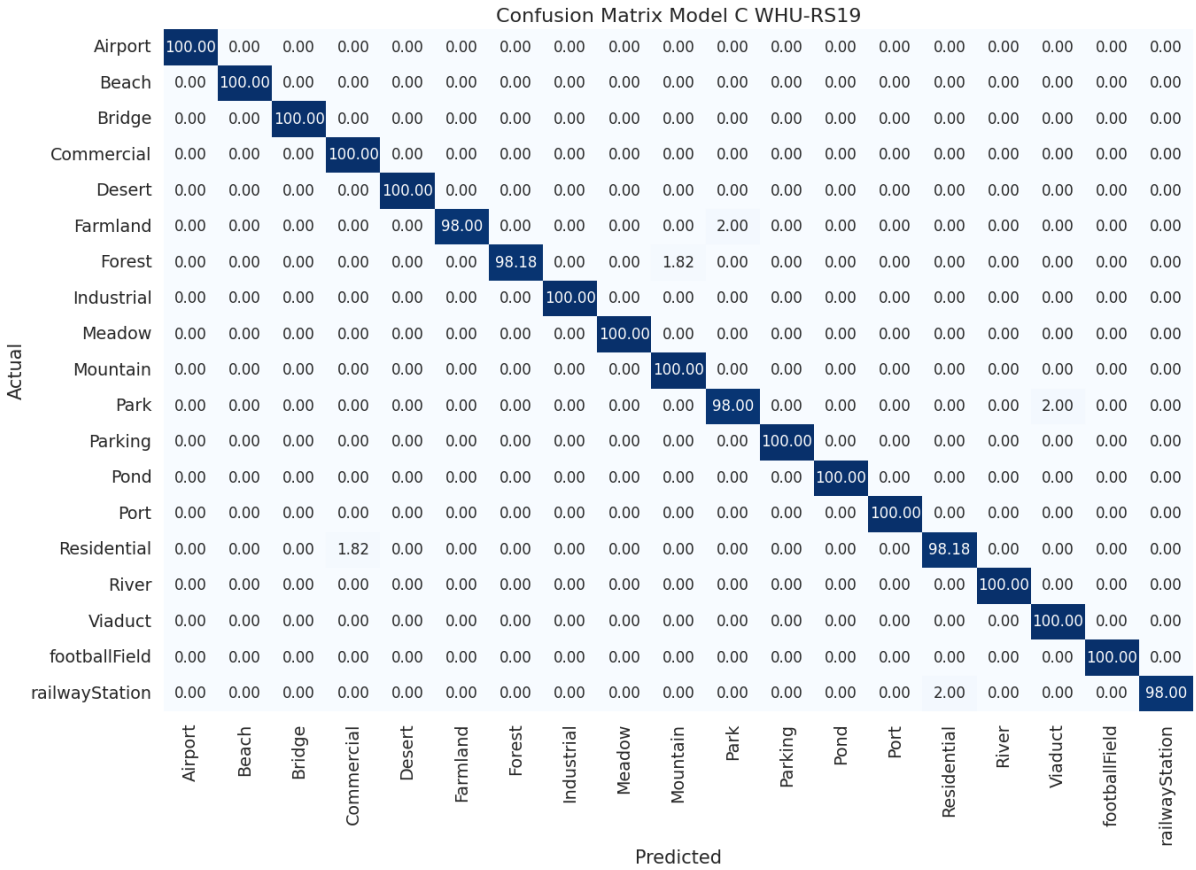

Figure 19 Confusion Matrix of Model C on WHU-RS19 dataset (in percentages)

In this paper, we will emphasize the comparison of this model with other models that also use ECANet to solve the problem of RSSC. The first is [10], which integrated the ECA module in each residual block of ResNet34 and obtained an accuracy improvement of 0.75% over the base model in AID (as shown in **Error! Not a valid bookmark self-reference.**). They also proved that ECANet can produce higher accuracy than SENet and CBAM. However, overall, our proposed model, which only adds ECANet after each stage instead of at each smallest unit block, can obtain higher accuracy (95.08%) than this model.

Table 7 Comparison of Accuracy in AID with Other Literature

Model	Accuracy (%)
VGG-VD16+SAFF [8]	93.83±0.28
VGG16-CapsNet [27]	94.74±0.17
Compact Deep Color Features [24]	94.30±0.24
VGG-VD16+MICP [26]	94.94±0.34
ResNet34 Standard [10]	93.35
ECA-ResNet34 Standard [10]	94.10
Model C (ours)	95.08±0.20

Table 8 Comparison of Accuracy in WHU-RS19 with Other Literature

Model	Accuracy (%)
EfficientNetB3+Attention2 [28]	99.47±0.20
Model C (ours)	99.50±0.31

The second model is RSCNet [17], which uses the basic ShuffleNet network. There is only one ECANet module added which is after the Conv5 block before the last GAP and Fully Connected layers. This model obtained slightly higher accuracy (99.05% on UC-Merced, shown in Table 6) than our proposed model, but it was trained with a very large number of epochs (200) and a fsmall batch size (16). In other words, it can be concluded that our proposed model is a trade-off between accuracy, complexity, and training time.

Model C obtained high Accuracy values of over 99% on the UC-Merced and WHU-RS19 datasets, while only 95% on the AID dataset. This shows that the proposed model is only able to achieve high Accuracy on small-sized datasets (UC-Merced and WHU-RS). It still struggles to achieve high Accuracy on large datasets, which have higher semantic variance and intra-class variance, and lower inter-class variance. Many solutions can be proposed to address this problem. From the model perspective, the model can be modified so that ECANet is inserted after each ConvNeXt block, rather than after each ConvNeXt stage which we used in this experiment. It should further improve the model's ability to reduce unimportant features from the feature map. However, it will also increase the training time.

From the data perspective, more image augmentation can be done to address the issue of high intra-class variance and low inter-class variance. There are cutting-edge image augmentation algorithms such as Cutout, CutMix and MixUp that can be employed to increase the accuracy of the RSSC model. For datasets with large image sizes such as AID, we could also consider using larger image sizes as input to CNN (e.g. 299×299), to dampen the effect of interpolation during the resize process.

4. Conclusions

Experiments have been conducted to perform Remote Sensing Scene Classification (RSSC) using the ConvNeXt-Tiny CNN deep learning model integrated with an Efficient Channel Attention (ECANet) Module and label smoothing regularization (LSR). ECANet is placed after each stage in ConvNeXt-Tiny so that in total four ECANets are added. We used the ImageNet pre-trained weights for the ConvNeXt-Tiny backbone. The activation function used to perform the final classification was softmax. The model was trained and tested on three datasets: UC-Merced, AID, and WHU-RS19. It has been shown that integrating ECANet in ConvNeXt-Tiny with LSR has increased the accuracy of the model in the RSSC task. The ConvNeXt-Tiny model with ECANet integration and LSR obtained an Accuracy of 99.00±0.41% in UC-Merced, 95.08±0.20% in AID, and 99.50±0.31% in the WHU-RS19, which are 0.38%, 0.73%, and 0.4% higher than its ConvNeXt-Tiny without ECANet and LSR counterpart, respectively. The model still has drawbacks in addressing low inter-class and high inter-class variance in large datasets, so we propose to optimize the model and perform more image augmentation in future research.

Acknowledgements

This work was empowered by an NVIDIA DGX-1 machine from Tokopedia-UI AI Center of Excellence

References

[1] G. Cheng, X. Xie, J. Han, L. Guo, and G.-S. Xia, "Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3735–3756, 2020, doi: 10.1109/JSTARS.2020.3005403.

[2] C. Patel, D. Labana, S. Pandya, K. Modi, H. Ghayvat, and M. Awais, "Histogram of Oriented Gradient-Based Fusion of Features for Human Action Recognition in Action Video Sequences," *Sensors*, vol. 20, p. 7299, Dec. 2020, doi: 10.3390/s20247299.

[3] Y. Li, H. Tang, W. Xie, and W. Luo, "Multidimensional Local Binary Pattern for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022, doi: 10.1109/TGRS.2021.3069505.

[4] W. Zhang, T. Zhou, C. xu, and M. Liu, "A SIFT-Like Feature Detector and Descriptor for Multibeam Sonar Imaging," *Journal of Sensors*, vol. 2021, Jul. 2021, doi: 10.1155/2021/8845814.

[5] N. Iqbal, R. Mumtaz, U. Shafi, and S. M. H. Zaidi, "Gray level co-occurrence matrix (GLCM) texture based crop classification using low altitude remote sensing platforms," *PeerJ Computer Science*, vol. 7, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:235453113

[6] R. Pires de Lima and K. Marfurt, "Convolutional Neural Network for Remote-Sensing Scene Classification: Transfer Learning Analysis," *Remote Sensing*, vol. 12, no. 1, 2020, doi: 10.3390/rs12010086.

[7] X. Zhao, J. Zhang, J. Tian, L. Zhuo, and J. Zhang, "Residual Dense Network Based on Channel-Spatial Attention for the

Scene Classification of a High-Resolution Remote Sensing Image," *Remote Sensing*, vol. 12, no. 11, 2020, doi: 10.3390/rs12111887.

[8] R. Cao, L. Fang, T. Lu, and N. He, "Self-Attention-Based Deep Feature Fusion for Remote Sensing Scene Classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 1, pp. 43–47, 2021, doi: 10.1109/LGRS.2020.2968550.

[9] J. Ji, T. Zhang, L. Jiang, W. Zhong, and H. Xiong, "Combining Multilevel Features for Remote Sensing Image Scene Classification With Attention Model," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 9, pp. 1647–1651, 2020, doi: 10.1109/LGRS.2019.2949253.

[10] Y. He, S. Zhou, and X. Quan, "Remote Sensing Image Scene Classification Based on ECA Attention Mechanism Convolutional Neural Network," in *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 2022, pp. 1265–1269. doi: 10.1109/ICCASIT55263.2022.9987089.

[11] Z. Zhao, J. Li, Z. Luo, J. Li, and C. Chen, "Remote Sensing Image Scene Classification Based on an Enhanced Attention Module," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 11, pp. 1926–1930, 2021, doi: 10.1109/LGRS.2020.3011405.

[12] R. Tombe and S. Viriri, "Remote Sensing Image Scene Classification: Advances and Open Challenges," *Geomatics*, vol. 3, no. 1, pp. 137–155, 2023, doi: 10.3390/geomatics3010007.

[13] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11966–11976. doi: 10.1109/CVPR52688.2022.01167.

[14] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2020, pp. 11531–11539. doi: 10.1109/CVPR42600.2020.01155.

[15] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions.," *J Big Data*, vol. 8, no. 1, p. 53, 2021, doi: 10.1186/s40537-021-00444-8.

[16] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, no. 4, p. 99, Mar. 2024, doi: 10.1007/s10462-024-10721-6.

[17] Z. Chen, J. Yang, Z. Feng, and L. Chen, "RSCNet: An Efficient Remote Sensing Scene Classification Model Based on Lightweight Convolution Neural Networks," *Electronics*, vol. 11, p. 3727, Nov. 2022, doi: 10.3390/electronics11223727.

[18] Z. Li, T. Gu, B. Li, W. Xu, X. He, and X. Hui, "ConvNeXt-Based Fine-Grained Image Classification and Bilinear Attention Mechanism Model," *Applied Sciences*, vol. 12, no. 18, 2022, doi: 10.3390/app12189016.

[19] X. Wang, "Improving Bag-of-Deep-Visual-Words Model via Combining Deep Features With Feature Difference Vectors," *IEEE Access*, vol. 10, pp. 35824–35834, 2022, doi: 10.1109/ACCESS.2022.3163256.

[20] Y. Long *et al.*, "On Creating Benchmark Dataset for Aerial Image Interpretation: Reviews, Guidances and Million-AID," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. PP, pp. 1–1, Apr. 2021, doi: 10.1109/JSTARS.2021.3070368.

[21] I. Kwong, F. Wong, T. Fung, E. Liu, R. Lee, and T. Ng, "A Multi-Stage Approach Combining Very High-Resolution Satellite Image, GIS Database and Post-Classification Modification Rules for Habitat Mapping in Hong Kong," *Remote Sensing*, vol. 14, p. 67, Dec. 2021, doi: 10.3390/rs14010067.

[22] S.-B. Chen, Q.-S. Wei, W.-Z. Wang, J. Tang, B. Luo, and Z.-Y. Wang, "Remote Sensing Scene Classification via Multi-Branch Local Attention Network," *IEEE Transactions on Image Processing*, vol. 31, pp. 99–109, 2022, doi: 10.1109/TIP.2021.3127851.

[23] M. Ismail, "A very high-resolution scene classification model using transfer deepCNNs based on saliency features," *Signal Image and Video Processing*, vol. 15, Jun. 2021, doi: 10.1007/s11760-020-01801-5.

[24] R. M. Anwer, F. S. Khan, and J. Laaksonen, "Compact Deep Color Features for Remote Sensing Scene Classification," *Neural Processing Letters*, vol. 53, no. 2, pp. 1523–1544, Apr. 2021, doi: 10.1007/s11063-021-10463-4.

[25] A. A. Aljabri, A. Alshanqiti, A. B. Alkhodre, A. Alzahem, and A. Hagag, "A Remote Sensing Scene Classification Model Based on EfficientNet-V2L Deep Neural Networks," *International Journal of Computer Science and Network Security*, vol. 22, no. 10, pp. 406–412, Oct. 2022.

[26] K. Qi, C. Yang, C. Hu, H. Zhai, Q. Guan, and S. Shen, "A multi-level improved circle pooling for scene classification of high-resolution remote sensing imagery," *Neurocomputing*, vol. 462, pp. 506–522, 2021, doi: https://doi.org/10.1016/j.neucom.2021.08.022.

[27] W. Zhang, P. Tang, and L. Zhao, "Remote Sensing Image Scene Classification Using CNN-CapsNet," *Remote Sensing*, vol. 11, no. 5, 2019, doi: 10.3390/rs11050494.

[28] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, "Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model With Attention," *IEEE Access*, vol. 9, pp. 14078–14094, 2021, doi: 10.1109/ACCESS.2021.3051085.

