IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 12, 2020, accepted December 31, 2020, date of publication January 12, 2021, date of current version January 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3051085

Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model With Attention

HAIKEL ALHICHRI, (Member, IEEE), ASMA S. ALSWAYED, (Member, IEEE), YAKOUB BAZI, (Senior Member, IEEE), NASSIM AMMOUR, (Member, IEEE), AND NAIF A. ALAJLAN, (Senior Member, IEEE)

Advanced Lab for Intelligent Systems Research (ALISR), Computer Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Corresponding author: Haikel Alhichri (hhichri@ksu.edu.sa)

This work was supported by the Researchers Supporting Project through the King Saud University, Riyadh, Saudi Arabia under Grant RSP-2020/69.

ABSTRACT Scene classification is a highly useful task in Remote Sensing (RS) applications. Many efforts have been made to improve the accuracy of RS scene classification. Scene classification is a challenging problem, especially for large datasets with tens of thousands of images with a large number of classes and taken under different circumstances. One problem that is observed in scene classification is the fact that for a given scene, only one part of it indicates which class it belongs to, whereas the other parts are either irrelevant or they actually tend to belong to another class. To address this issue, this paper proposes a deep attention Convolutional Neural Network (CNN) for scene classification in remote sensing. CNN models use successive convolutional layers to learn feature maps from larger and larger regions (or receptive fields) of the scene. The attention mechanism computes a new feature map as a weighted average of these original feature maps. In particular, we propose a solution, named EfficientNet-B3-Attn-2, based on the pre-trained EfficientNet-B3 CNN enhanced with an attention mechanism. A dedicated branch is added to layer 262 of the network, to compute the required weights. These weights are learned automatically by training the whole CNN model end-to-end using the backpropagation algorithm. In this way, the network learns to emphasize important regions of the scene and suppress the regions that are irrelevant to the classification. We tested the proposed EfficientNet-B3-Attn-2 on six popular remote sensing datasets, namely UC Merced, KSA, OPTIMAL-31, RSSCN7, WHU-RS19, and AID datasets, showing its strong capabilities in classifying RS scenes.

INDEX TERMS Remote sensing, scene classification, EfficientNet-B3, convolutional neural networks (CNNs), attention mechanisms.

I. INTRODUCTION

Recent years have witnessed a rapid development in remote sensing (RS) technologies [1]–[3]. The advancement in monitoring capabilities and the growing number of remote sensing platforms, has permitted to obtain a large number of geographical images over the earth surface with different spatial, spectral and temporal resolution [4]. Analyzing these images plays an important social and economic role, as they represent a valuable source of information for decision making in various applications, such as natural disaster detection [5], agricultural survey, urban planning [6], natural resources monitoring, weather forecasting, land-cover/land-use classification and geographic space object detection/retrieval [7], [8].

Scene classification in remote sensing can be described as a method that focuses on classifying RS scenes into a set of classes according to the contents of that scene. A large amount of works that deals with this problem is published in the RS literature [9]–[29]. Recent state-of-the-art works use Convolutional Neural Networks (CNN) models to learn rich feature representation of RS scenes and classify them. Most of these works learn feature that represent the RS scene as a whole. However, oftentimes only one part of that image is important in telling which class it belongs to, and the other parts are irrelevant or belong to another class. Thus the parts of the scene that are irrelevant and/or belong to other classes, may actually confuse the algorithm. This observation

The associate editor coordinating the review of this manuscript and approving it for publication was Jeon Gwanggil.

motivates the proposition of deep learning models that incorporate an attention mechanism. These mechanisms make the model learn how to focus on the parts of the scene that matter most in the classification process, which should improve the classification accuracy.

A typical RS scene shown in Fig. 1 belonging to the "Storage tanks" class, we can see that the part of the scene in the red rectangle is the most relevant, whereas the background contains many clutter that may even belong to other classes (for instance the class of "grass" or "trees" in the Fig. 1).

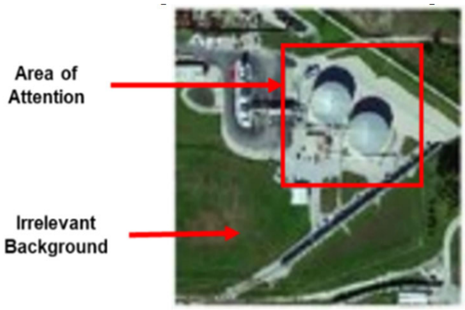

FIGURE 1. Sample RS scene from the "storage tanks" class. We can observe that some parts are irrelevant and belong to other classes (such as "grass").

The attention mechanism (the idea of focusing on specific parts of the input) has been applied in deep learning for speech recognition [30], Natural Language processing [31], multimodal reasoning and matching [32], object detection [33], and image recognition [34]–[36]. In remote sensing, some works that use attention are proposed for RS object detection [37], RS image segmentation [38], [39], and RS scene classification [40]–[49].

Wang *et al.* [40] presented the first work that incorporates attention in RS scene classification, where they propose the Attention Recurrent Convolutional Network (ARCNet) for scene classification. ARCNet learns to adaptively select a series of attention regions and then process then sequentially to generate powerful predictions. They also design a novel recurrent attention structure to squeeze high-level semantic and spatial features into several simplex vectors for the reduction of learning parameters.

The authors in [41] introduces attention-based weighting scheme into ensemble learning. Their method called convolutional attention in ensemble (CAE), transfers the knowledge contained in base classifiers into the final classifier using convolutional attention models.

Another work in [43] propose a novel Saliency Dual Attention Residual Network (SDAResNet) to extract both cross-channel and spatial saliency information for scene classification of RSI. More specifically, spatial attention is embedded in low-level feature to emphasize saliency location information and suppress background information, and channel attention is integrated to high-level features to extract saliency meaningful information.

Alswayed *et al.* [45] propose a deep attention model based on the pre-trained SqueezeNet CNN for RS scene classification. They introduce a separate branch to the network that implements an attention mechanism and learns learn the best weights for features learned in the main branch. Feature vectors that are assigned a higher weight indicate that the network has given more attention to the receptive field in the scene corresponding to that feature vector.

The authors in [46] propose an attention mechanism-based convolutional neural network with multiaugmented schemes to improve the RS scene classification problem. An augmentation operation over attention mechanism feature maps are used to force the model to capture class-specific features and eliminate redundant information and push the model to capture discriminative regions as much as possible, instead of using all global information without favor.

The work in [47] presents another attention–base method for RS scene classification that can discriminate the crucial information from the complex scene content. The method is based on the DenseNet CNN model as a back bone and is called channel-attention-based DenseNet (CAD). DenseNet CNN can extract spatial features at multiple scales and correlate with each other. Then a channel attention mechanism is introduced to strengthen the weights of the important feature channels adaptively and to suppress the secondary feature channels.

More recently, the authors [48] propose a dual attention-aware network for RS scene classification. Again, they use two kinds of attention modules, channel and spatial attentions. The outputs of two attention modules are finally integrated as the attention-aware feature representation for improving classification performance. The classification network is composed of three subnetworks, which are trained by certain scaled regions separately, then their feature outputs are fused together before final classification.

Another work [49] presents a method for utilizing the attention mechanism to localize multiscale discriminative regions of the RS scene images and combining features learned from the localized regions.

To understand why attention works in deep learning, we have to think about what a neural network really is: *a function approximator*. Its ability to approximate different classes of functions depends on its architecture. A typical neural net is implemented as a chain of matrix multiplications and element-wise non-linearities, where elements of the input or feature vectors interact with each other only by addition. Attention mechanisms compute a mask which is used to multiply features. This seemingly innocent extension has profound implications: suddenly, the space of functions that can be well approximated by a neural net is vastly expanded, making entirely new use-cases possible

In this work, we propose a deep learning method for RS scene classification based on the new EfficientNet CNN model combined with an attention mechanism. Specially, our proposed CNN model, named EfficientNet-B3-Attn, is a modified version of the EfficientNet-B3 CNN, where a branch is added to learn a set of weights that are used to combine convolutional features in intermediate layers of the network. The suitable intermediate layer is determined from experimental results. The main contributions of the paper include the following points:

1) We propose a method to classify RS scenes based on the EfficientNet-B3 CNN model and the attention mechanism.
2) The attention mechanism is applied at the feature level as a secondary branch attached at the end of layer 262 of the original EfficientNet-B3 model. The suitable layer number is determined experimentally. Then the outputs of both branches are averages to produce the final prediction probabilities. We call this novel proposed model EfficientNet-B3-Attn-2.
3) We test the proposed EfficientNet-B3-Attn-2 model it on six RS scene datasets to evaluate its performance.

The rest of the paper is organized as follows; Section II describes the family of EfficientNet CNN models and the proposed methodology in Section III. Then, in section IV, we present the datasets used and experimental results. Finally, conclusions and possible future research is presented in section V.

II. THE FAMILY OF EFFICIENTNET MODELS

Recently, Tan and Le [50] studied the relationship between width and depth of CNN models and came up with an efficient way to design CNN models that have less parameters, yet they provide better classification accuracy. They called them EfficientNet CNN models and in their original paper they proposed seven such models which they named EfficientNetB0 to EfficientNetB7. Tan and Le [50] show that the EfficientNet CNN models outperform all previous models both in term of the number of parameters and Top-1 accuracy when applied to the ImageNet dataset [51].

The EfficientNet family is based on a new method for scaling up CNN models. It uses a simple greatly effective compound coefficient. Differently from traditional methods that scale dimensions of networks, such as width, depth, and resolution, EfficientNet scales each dimension with a fixed set of scaling coefficients uniformly. Practically, scaling individual dimensions improves model performance, however balancing all dimensions of the network with respect to the available resources effectively improves the whole performance.

Compared to other models achieving similar ImageNet accuracy, EfficientNet is much smaller. For example, the ResNet50 model as you can see in Keras application has 23,534,592 parameters in total, and yet, it still underperforms the smallest EfficientNet (called EffecientNet-B0), which only has 5,330,564 parameters in total. In this work, we present an efficient method based on the EfficientNet-B3 CNN model. We select this particular variant of the EfficientNet family because it provides a good compromise between computational resources and accuracy. The same ideas we present here can be applied to the other more powerful variants.

Mobile inverted bottleneck convolution (MBConv) is the main building block of EfficientNet model family. MBConv is based on concepts borrowed from the MobileNet models [52]. One main idea is using depthwise separable convolutions, which consist of a depth wise and a pointwise convolution layers after one another. Then two more ideas are borrowed from MobileNet-V2 (which is a second improved version of MobileNet) including; 1) inverted residual connections, and 2) Linear bottlenecks.

Fig. 2 presents and illustration of inverted residual blocks. In the original residual blocks defined ResidualNets [53], the skip connections exist between layers with wide number of channels (64 in Fig. 2a).

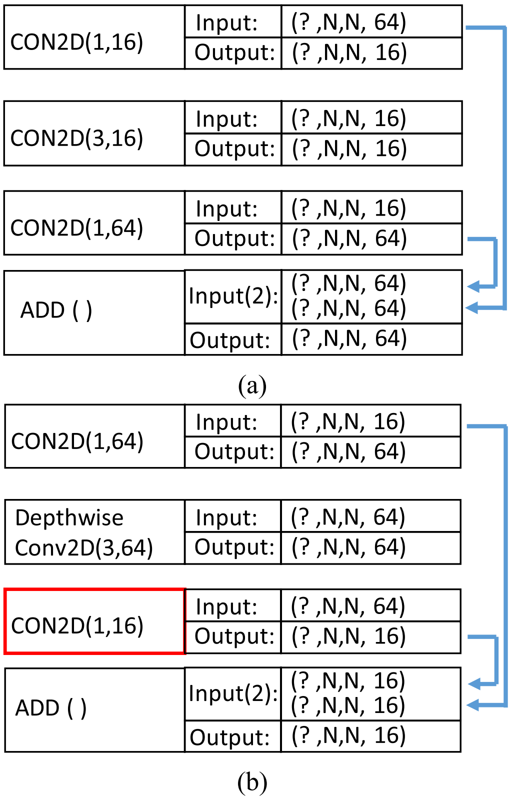

FIGURE 2. Inverted residual block example, (a) regular residual block where channel size changes from 64 → 16 → 64, (b) inverted residual block where channels change from 16 → 64 → 16.

Inside the residual block the number of channels are reduced or squeezed to 16, so that the number of parameters required by the 3 × 3 convolutions in the next layer is also reduced. In the inverted residual block shown in Fig. 2b, the sizes of the connected channels are inverted, so that now the skip connections are taking place between narrower layers with small number of channels. This explains the reason for the name inverted residual blocks. In this latter type and even though the number of channels in the layer inside the block increases to 64, the number of parameters is actually lower

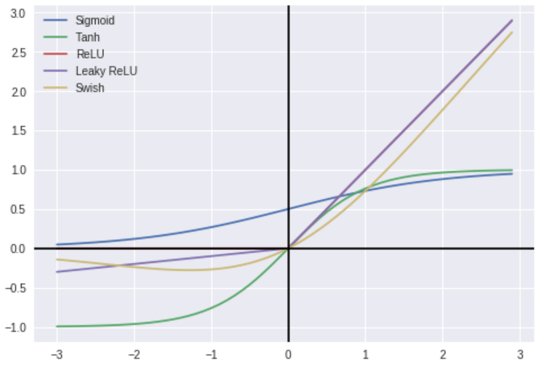

FIGURE 3. The new Swish activation function [54]. Compared to ReU and LeakyReLU, the swish has a similar shape but is smoother.

compared to the original residual block of ResNet, because we use depthwise convolutions.

The second idea in MobileNetV2 is linear bottlenecks, which means that we use linear activation function for the layer highlighted in red color in Fig. 2b. This is called a bottleneck layer because the number of channels is squeezed at these locations of the network. The authors who proposed the MobileNetV2 CNN argue that the ReLU activation function that is commonly used in CNN architectures does not work well with inverted residual blocks because it discards values that are smaller than zero. Using linear activation function for the layer with reduced channels (bottleneck channel) produced better performance.

Additionally, this network uses a new activation function called Swish instead of the ReLU activation function. As shown in Fig. 3, the Swish activation function is similar in shape to the ReLU and LeakyReLU functions and hence shares some of their good performance advantages. However, unlike these two, it is a smoother activation function.

Formally, the Swish function is defined in Equation (1):

$$f_{Swish}(x) = \frac{x}{1+e^{-\beta x}} \tag{1}$$

where $\beta \geq 0$ is a parameter that can be learned during training of the CNN model. Note, if $\beta = 0, f_{Swish}$ becomes the linear activation function and as $\beta \rightarrow \infty, f_{Swish}$ looks more and more like the ReLU function except it is smoother as shown in Fig. 3.

The effectiveness of the model scaling idea that is mentioned earlier, depends strongly on the baseline network. To this end, a new baseline network is created by using the automatic machine learning (AutoML) MNAS framework, which automatically searches for a CNN model that optimizes both precision and efficiency (in FLOPS). This baseline network is called EfficientNet-B0 and its main architecture is shown in Fig. 4.

The first observation is that this bassline model is composed of repeated MBConv1, MBConv3, and MBConv6 blocks. These are basically different types of MBConv block. The second observation is that inside each block the number of channels is increased or expanded (through a larger number of filters). The third observation is the inverted residual connections which are taking place between the narrow layers of the model.

The authors in [50] also included the concept of squeeze-and-excitation (SE) in the MBConv blocks, which contributes to further performance improvements. The SE idea is illustrated by Fig. 5.

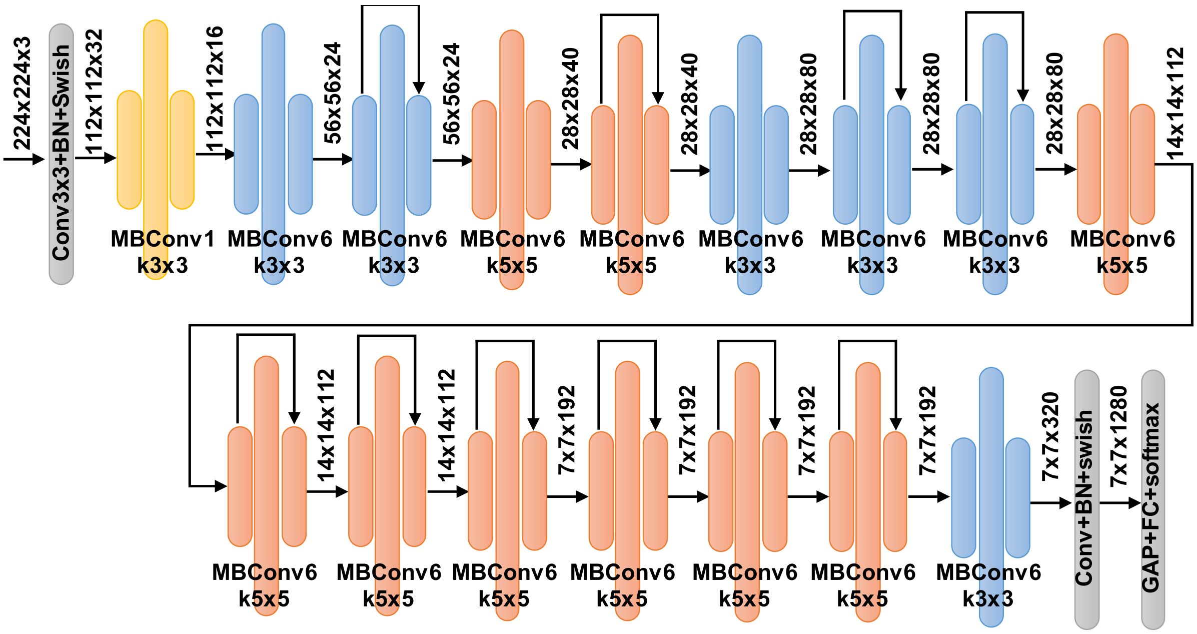

FIGURE 4. The EffecientNet-B0 general architecture.

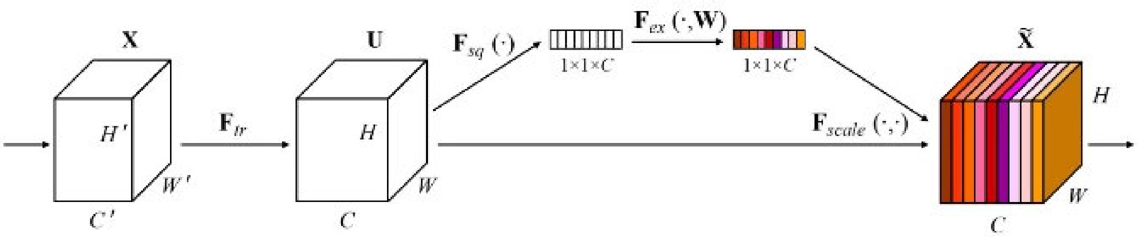

FIGURE 5. Illustration of the squeeze-and-excitation concept [55].

Recall that the output of a convolutional layer consists of set of channels which are defined by the number of filters parameter. Typically, these channels are given equal weight in future operations. The SE block is a technique that gives a different weightage to each channel instead of treating them all equally. The upper branch in Fig. 5 learns a set of weights (highlighted by colors) equal to the number of channels C, and then the original feature channels are scaled by these weights. The SE block gives the output of shape (1 x 1 x channels) which specifies the weightage for each channel and the great thing is that again these weightage values are learned during training like other parameters.

Finally, we present in Fig. 6 an example MBConv block which takes as input a feature map of size (56 x 56 x 24) and includes all the concepts mentioned above, including the 1) depthwise separable convolutions, 2) inverted residual blocks, 3) linear bottlenecks, 4) Swish activation functions, and 5) the SE block. There are also several types of MBConv blocks. The particular type shown in Fig. 6 is called MBConv6. Another type, shown in Fig. 7, is called MBConv1 which is used at the beginning of the Effeicnet-Net models. In addition, each type can have several variants depending on the filter size used in the convolutional layers inside the block (which can be 3 × 3 or 5 × 5), and depending on whether the block contains an inverted residual connection or not.

The other EfficientNet CNN models are defined through the model scaling idea and are, hence, deeper and wider. For example, EffecientNet-B3 model is shown in Fig. 8, where IRC means that the MBConv block uses an inverted residual connection. Similar, to EffeicienNet-B0 it used MBConv1 and MBConv6 modules. Not all modules use inverted residual connection (IRC). The modules that use this type of connection are indicated by the acronym IRC. The other modules cannot have this connection because the input size is not the same as its output size and thus cannot perform an add operation

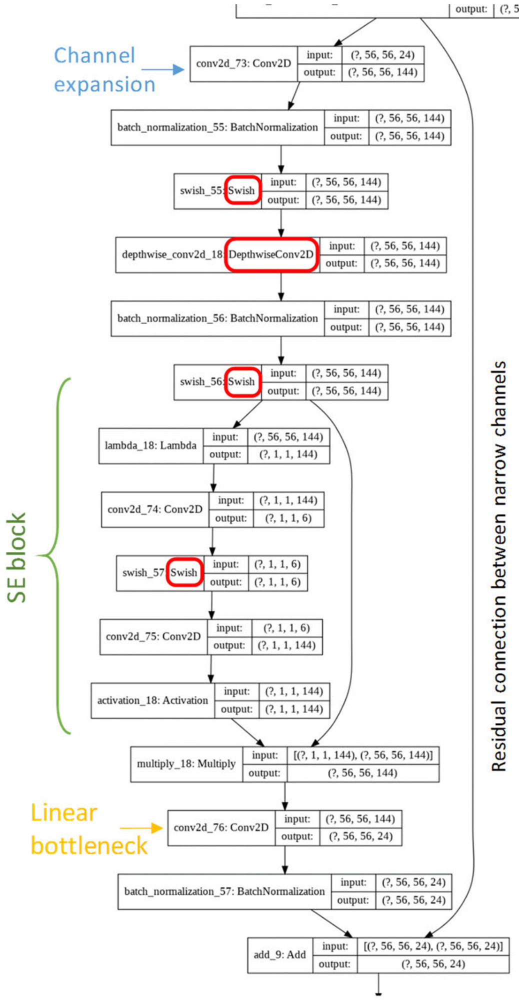

FIGURE 6. Illustration of MBConv6 with Squeeze-and-Excitation block and inverted residual connection 24 → 144 → 24.

III. PROPOSED METHODOLOGY

Typically, a CNN model involves several convolutional layers that operate on an image consecutively as shown in Fig. 9. The convolutional layers are intertwined with other types of layers such as pooling layers, normalization layers, and activation function layers which help the model approximate non-linear functions. However, for simplicity we will ignore these model details because they are not relevant to the next discussion.

Notice that the neurons in the first convolutional layer capture the features in a small area in the image. If the size of the filters used is 3 × 3 than that will be the size of this area. We call this area the receptive field of that neuron in the image. In the next layer of the CNN each neuron is convolved with the same 3 × 3 area in the previous layer but

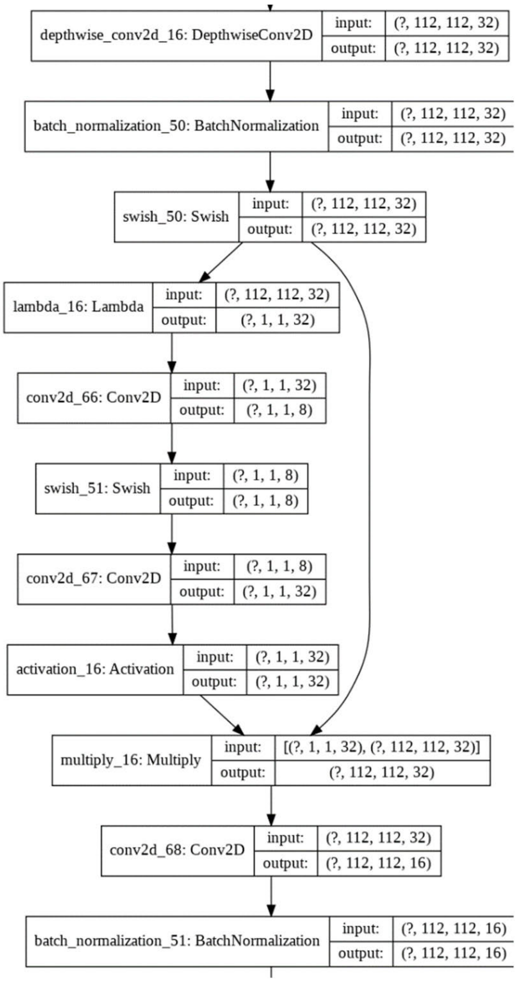

FIGURE 7. Illustration of the MBConv1 block type.

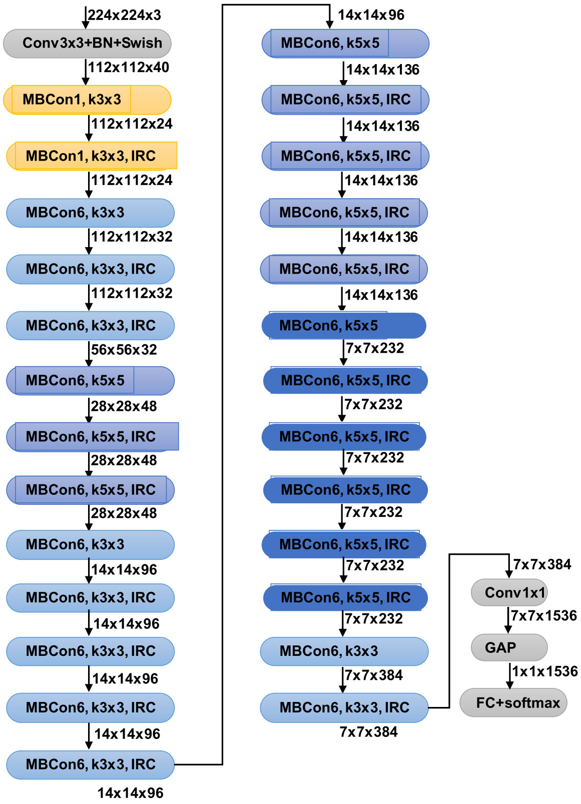

FIGURE 8. Illustration of the EffecientNet-B3 architecture. (10,3646 million weights). IRC means there is inverted residual connection.

that translates to a large receptive field in the input image. As we go deeper and deeper into the network each neuron corresponds to a larger and larger receptive field.

The vector of neurons along the third dimension is a learned feature vector representing the receptive filed in the image. Thus, in Fig. 9, the yellow feature vector of the last convolutional layer may represent the receptive field shown in the dashed red square. The same goes for the other feature vectors (blue, green, and red in the Fig. 9), they represent different receptive fields in the image. These different receptive fields represent different regions in the image. Thus, we can apply attention at this level by a weighted combination of the feature vectors. We can make the model learn to pay attention to important regions of the image, by learning the relative weights we assign to their corresponding feature vectors.

At the end of the CNN model, the last features extracted are converted into one vector through two options 1) a flatten operation where the individual vectors are stacked on top of each other (see Fig. 9a), or 2) a pooling operation where the feature vectors are added, multiplied, averaged, or other operations. For example, Global Average Pooling (GAP) is used in Fig. 9b, which is found to be more robust in practice.

However, GAP performs a simple average with equal weights, which does not give any special attention to particular receptive fields or regions of the input image. Thus, the GAP operation performs the operation shown in the equation below:

$$\mathrm{L}_j = GAP\left(\mathrm{L}_{j-1}\right) = \frac{1}{M \times M}\sum\nolimits_{i=0}^{M\times M} \mathrm{F}_i \tag{2}$$

where L_j and L_{j-1} are the input and output layers of the GAP function. F_i is the feature vectors contained in layer L_j, and MxM is the number of feature vectors (2×2 in Fig. 9). In order to implement attention, we need to compute a weighted average as follows:

$$\mathrm{L}_j = GAP\_ATN(\mathrm{L}_{j-1}) = \sum\nolimits_{i=1}^{M\times M} \mathrm{w}_i\mathrm{F}_i \tag{3}$$

where w_i are weights to be learned by the model automatically. A dedicated branch of network layers is added to the model to learn the most suitable weights from each image that focus attention on its relevant regions.

Formally, the attention mechanism equips a neural network with the ability to focus on a subset of the feature vectors by giving them different weights. Let $\boldsymbol{x} \in \mathcal{R}^{\boldsymbol{d}}$ be an input vector, $z \in \mathcal{R}^{\boldsymbol{k}}$ a feature map, $\boldsymbol{a} \in [\mathbf{0}, \mathbf{1}]^{\boldsymbol{k}}$ an attention vector, $\boldsymbol{g} \in \mathcal{R}^{\boldsymbol{k}}$ an attention glimpse and $\boldsymbol{f}_{\emptyset}(\boldsymbol{x})$ an attention network, that can be a branch in the same network, with parameters Ø.

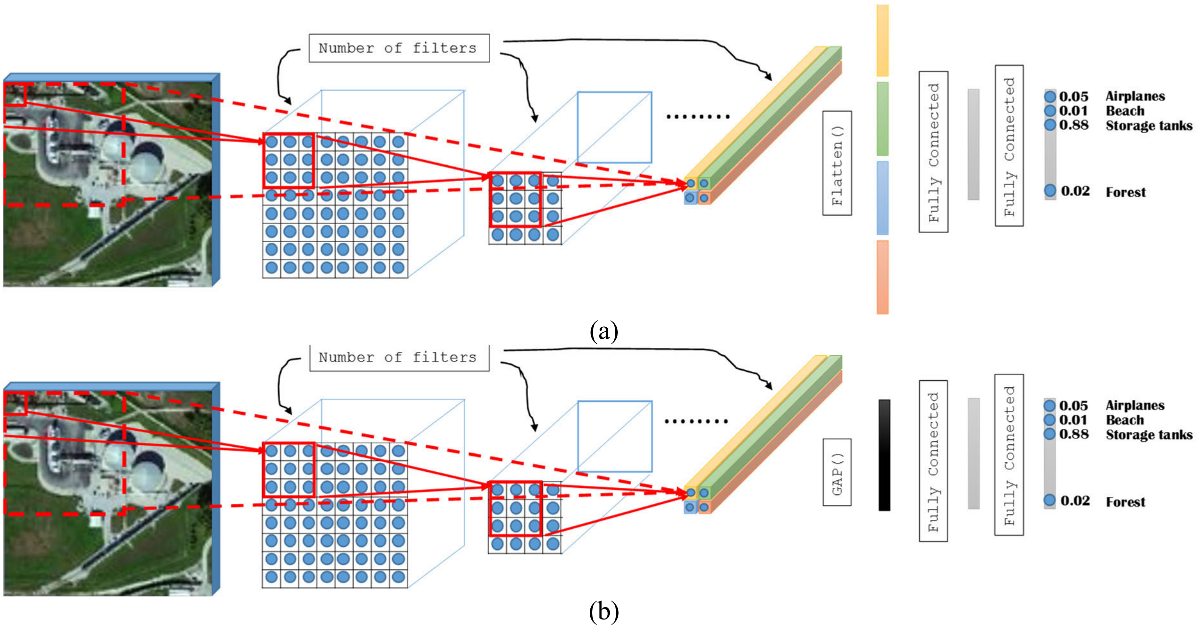

FIGURE 9. Typical CNN architecture with two options after the last convolutional layer (a) The last feature vectors are flattened into one large vector, (b) The last feature vectors are averaged together.

Typically, attention is implemented as:

$$a = f_{\emptyset}(x)$$
$$g = a \odot z \tag{4}$$

where $\odot$ is element-wise multiplication. The attention weights a take values between zero and one, i.e. $\boldsymbol{a} \in [\mathbf{0}, \mathbf{1}]^{\boldsymbol{k}}$. If a is either zero or one, then this is called hard attention as opposed to the soft attention where the weights take values between zero and one. In our work here, we consider the soft attention approach.

A. ADDING THE ATTENTION MECHANISM

The attention layers are illustrated in Fig 10. This can be applied to any feature map in the CNN model. Let us assume the size of the input feature map to be N × N × C, where N × N is the 2D map size and C is the number of channels. The attention module starts by squeezing the feature map using two consecutive convolutional layers so that the size is N × N × 16. Then it uses a locallyConneced2D layer followed by a sigmoid activation function to learn N × N weights. Then another convolutional layer is used to replicate the weights across the channel dimension C times. It is important to note here that this layer is followed by a linear activation function, which means the weights can take on a wide range of values.

However, after averaging the new feature map with attention into one feature vector of length C, we scale the result through division by the average weight vector. This ensures that the final operation behaves like a weighted averaging where values are kept comparable in magnitude to the original feature vectors.

Since the large EfficientNet models achieve better accuracy compared to smaller models, we initially planned to use the EffecientNet-B7 model variant. However, constrained by the computational resources available to us, we opted for EffecientNet-B3 variant as a good compromise between accuracy and parameter count.

We first add the attention layers at the top of the model, i.e. we remove the last two layers in Fig. 8, and replace them with the attention module illustrated by Fig. 10. Consequently, we now have two versions one without attention (Fig. 8) and the other with attention. We call these two

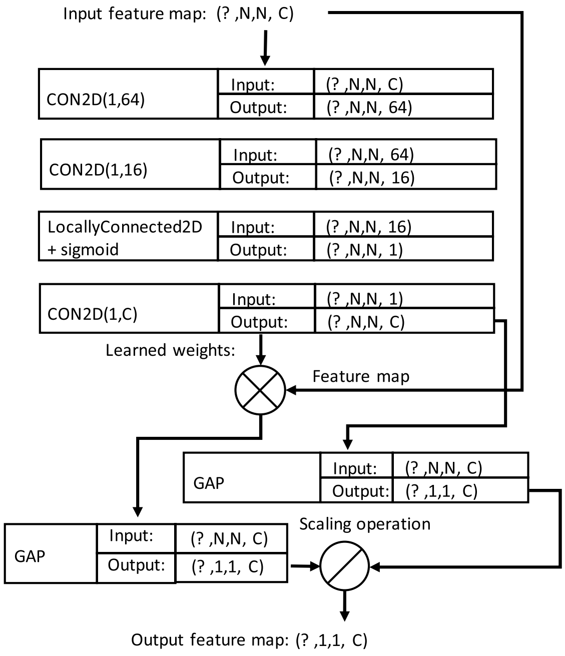

FIGURE 10. Details of the proposed attention module.

versions EffecientNet-B3-Basic and EfficientNet-B3-Attn-1, respectively.

However, another idea is to incorporate the attention module at lower convolutional layers because the higher layers' features represent very large receptive fields with highly overlapping regions, which means the attention mechanism may not be effective with these features. Thus we also propose to investigate other options as shown in Fig. 11, where the attention module is added as a second branch in the network starting from different MBConv blocks.

As can be seen in Fig. 11, we investigate several positions for the attention branch including MBConv blocks 9,14, 19, 25, and the last 27th block. In other words, the model now has two separate branches. This also means that the model has two outputs which must be optimized jointly. The final proposed model called EfficientNet-B3-Attn-2, is shown in Fig. 12, where the attention module is connected to Block 19.

B. MODEL OPTIMIZATION

Typically, deep models are trained end-to-end using backpropagation technique minimizing the so-called cross-entropy loss:

$$E = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{C} 1\left(y_{ik} = k\right) \ln\left(\frac{\exp\left(\left(w_k^{out}\right)^T h_k^{out}\right)}{\sum_{j=1}^{C}\exp\left(\left(w_j^{out}\right)^T h_j^{out}\right)}\right) \tag{5}$$

where n is the number of training samples, C is the number of classes, y_{ik} is the prediction probability for sample i and class k, h_k^{out} are the output of the last hidden layer and w_k^{out} are the weight matrix from that hidden layer to the output layer. The formulation $1(\cdot)$ is an indicator function that takes 1 if the statement is true, otherwise it takes 0

For our proposed EffecientNet-B3-Attn-2 where we have two outputs, there will two cross-entropy errors E_1 and E_2 and they need to be minimized jointly. We do this by optimizing a weighted sum of the two errors:

$$E = \gamma_1 E_1 + \gamma_2 E_2 \tag{6}$$

where $\gamma_1$and $\gamma_2$are positive hyper-parameters that controls the trade-off between the output of the two branches. We rewrite Equation (12) as:

$$E = \gamma_1 (E_1 + \frac{\gamma_2}{\gamma_1} E_2) \tag{7}$$

Thus only one parameter $\lambda = \frac{\gamma_2}{\gamma_1}$ is needed for this equation because scaling the loss E by a positive parameter γ_1 does not affect the minimization problem. In our work, we set $\lambda = 1$, because there should be no preference to one output over the other.

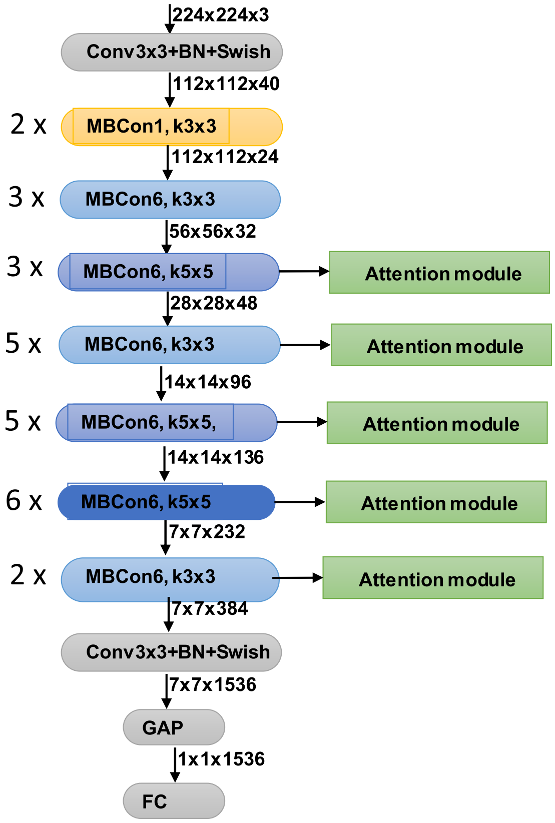

FIGURE 11. EffecientNet-B3-Attn-2 (10,3646 million weights): attention module incorporated at some intermediate layer of the model.

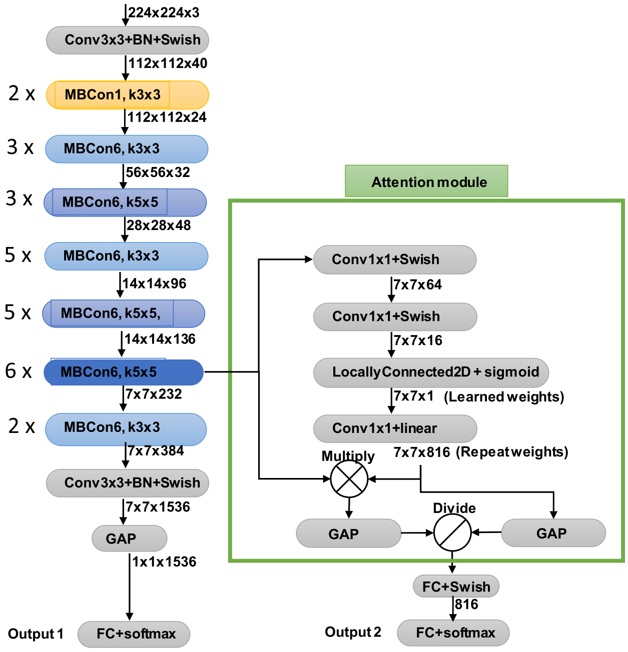

FIGURE 12. EffecientNet-B3-Attn-2: Attention mechanism is added to the original EfficientNet-B3 model as second branch connected to Block 19.

IV. EXPERIMENTAL RESULTS

In this section, we present extensive experimental work to show the capabilities of the proposed solution. In total we used six datasets to test our method. The datasets are the University of California (UC) Merced dataset [56], the Kingdom of Saud Arabia (KSA) dataset [57], [58],

TABLE 1. Summary of dataset properties.

Dataset	No. of Images	No. of classes	Images per class	Image size	Resol. m/pixel
UC Merced	2100	21	100	256x256	0.3
KSA	3250	13	250	256x256	0.5
RSSCN7	2800	7	400	400x400	N/A
Optimal31	1860	31	60	256x256	0.5
Whurs19	1005	19	50	600x600	0.5
AID	10000	30	Varies	600x600	0.5

the RSSCN7 dataset [10], the OPTMIAL-31 dataset [40], the WHU-RS19 dataset [15], and the last and largest one is the Aerial Image Datasets (AID) dataset [59].

Table 1 gives a summary of the properties of these three datasets. We have resized all these datasets to 256×256 pixels and as a contribution to the RS community, we made all the resized datasets available online [60].

A. EXPERIMENTAL SETUP

The image sizes vary across datasets, however image sizes 256 × 256 is more common. Thus, in this work, we resize all datasets to 256 × 256 as is done by most other works to reduce training times and also for consistency.

We use three train-test splits, where the dataset is randomly divided into two subsets one used for training and the other for testing. In particular, we have tested the following splits 20%-80%, 50%-50% and 80%-20%, where the first value is the percentage of training data while the second one is the percentage of the testing data. We perform these splits randomly five times and then consider the average classification results.

To evaluate the overall performance of the proposed method, we use the overall accuracy (OA), which is the fraction of correctly classified samples in relation to all samples for testing:

$$OA = \frac{\sum_{i=0}^{C} \mathrm{n}_{ii}}{|T|} \tag{8}$$

where C is the number of classes and |T| is the total number of test samples.

All experiments were conducted on the Colab environment of Google using the Tensorflow machine learning library written in python. To find the optimum weights of the network we use the backpropagation algorithm with a batch gradient optimization method. In particular, we use the advanced optimization algorithm Adam with all of its parameters set to their defaults values except for learning rate. We set the learning rate parameter to 0.001 during the first 15 epochs, then for the next 15 epochs it is reduced to 0.0001. Finally, due to memory limitations of this platform, the model is trained in batches of 32 images at a time.

B. RESULTS USING THE OPTIMAL-31 MODEL

In the first set of experiments we study the proposed method using the OPTIMAL-31 dataset. OPTIMAL-31 dataset is one of most challenging dataset because of the high data variability in terms of different sensors, different scales, and so on. In Table 2, we report the results of the EffecientNet-B3-Basic (without attention) and EffecientNet-B3-Attn-1 (with attention) as described in Section III A.

TABLE 2. Results of efficientnet-b3 basic model versus efficientnet-b3-attn-1 using the optimal-31 dataset.

Train/Test split	EfficientNet-B3-Basic	EfficientNet-B3-Attn-1
20%-80%	83.18+0.21	83.03+0.05
50%-50%	90.80±0.84	90.03±0.24
80%-20%	94.63+0.30	92.33+0.85
Total parameters	10,831,175	10,939,351
Trainable	10,743,879	10,847,447

The results in Table 2, Indicate that the initial proposed model EfficientNet-B3-Attn-1 is not suitable. Applying attention at the last layer reduced the accuracy of the model. This can be explained by the fact that each neuron has a large receptive field that may even cover the whole image. Thus applying attention will not help focus on particular parts of the image.

In the second set of experiments, we investigate the EfficientNet-B3-Attn-2 with an attention branch added to intermediate layers as illustrated in Fig. 12. First, we investigate the effect of adding the attention branch to different intermediate layers and show the results in Table 3. In particular, we test layers located at the end of the last seven MBConv blocks of the EfficientNet-B3 model. From Table 3, we can see that incorporating the attention mechanism at layer number 262, i.e., at the start of the 19th MBConv block, provides the best classification accuracy.

TABLE 3. Results using the OPTIMAL-31 dataset of EfficientNet-B3-Attn-2 model where the attention mechanism is added at different locations.

	Train-Test split		
location of attention branch	20%-80%	50%-50%	80%-20%
70	84.14%	88.97%	94.09%
114	84.48%	89.68%	94.35%
188	84.94%	91.08%	93.82%
262	**87.07%**	**92.37%**	**96.51%**
351	85.28%	90.97%	94.89%
380	83.03%	90.03%	92.33%

For illustration of the effectiveness of the method, Fig. 13 shows the attention weights learned by the EffecientNet-B3-Attn-2 model (Brighter colors indicate higher attention).

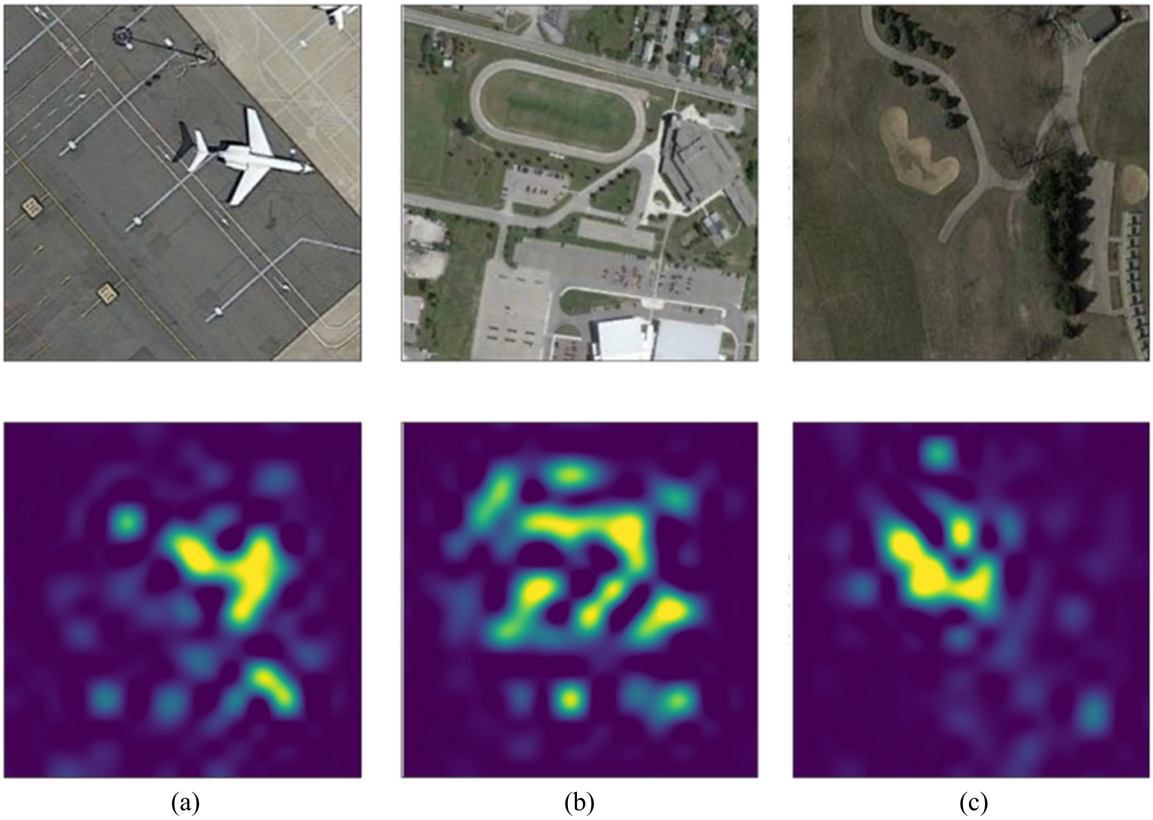

FIGURE 13. Improved Images and their attention map for OPTIMAL-31 dataset with a 20%-80% train-test split using. Improvements from incorrect class to correct class after attention are (a) from Airport to Airplane, (b) from Basketball court to Playground, and (c) from Baseball field to Golf field.

Here we show samples of the images from the OPTIMAL-31 dataset, that have their classification corrected using the EffecientNet-B3-Attn-2 model as opposed to the EffecientNet-B3-Basic model. It is clear that our proposed model learns to pay attention to important information in regions corresponding to true labels and aggregate features from these important regions.

C. RESULTS USING ALL DATASETS

In this section we evaluate the performance the proposed EffecientNet-B3-Attn-2 model using all of the six datasets. We present the results in Table 4 for all dataset and for three train-test splits. As can be seen, the proposed method presents a significant improvement in terms of overall accuracy compared to the model that does not use the attention.

D. COMPARISON TO STATE-OF-THE-ART

In this section, we summarize the results obtained for the proposed EffecientNet-B3-Attn-2 model and compared it with some state-of-the-art methods on six public RS scene datasets. The experiments are repeated five times to reduce the influence of the randomness for a reliable result. The mean and standard deviation of overall accuracies on the testing sets from each run were reported. In these experiments, we used the following different train/test splits 20%-80%, 50%-50%, and 80%-20% for all datasets except the WHU-RS19 and AID datasets. For the WHU-RS19 we used 40%-60%, 60%-40%, and 80%-20%, because that is what is used in the literature. As for AID,

TABLE 4. Summary of the overall accuracy of the proposed EffecientNet-B3-Attn-2 for all datasets.

Training ratio		EfficientNetB3 with no attention	EffecientNet-B3-Attn-2
UC Merced	20%	94.11±0.15	95.60±0.31
	50%	97.63±0.06	97.90±0.36
	80%	98.73±0.20	99.21±0.22
KSA	20%	95.77±0.12	96.38±0.18
	50%	97.29±0.06	97.65±0.17
	80%	97.68±0.32	97.99±0.27
RSSCN7	20%	92.06±0.39	93.30±0.19
	50%	94.39±0.10	96.17±0.23
	80%	95.18±0.21	96.89±0.20
OPTIMAL-31	20%	84.02±0.93	84.43±0.56
	50%	90.80±0.84	91.80±0.54
	80%	94.76±0.26	95.86±0.22
WHU-RS19	40%	97.28±0.24	98.60±0.40
	60%	97.68±0.10	98.68±0.93
	80%	98.95±0.20	99.47±0.20
AID	20%	93.43+0.33	95.37+0.41
	50%	94.45+0.76	96.56+0.12

TABLE 5. Comparison of EffecientNet-B3-Attn-2 model with the state-of-the-art performance on the UC Merced dataset.

Method	Year	Training ratio		
		20%	50%	80%
Fine-tuning GoogLeNet [61]	2015			97.10
VGG16-G-IFK [15]	2015			98.49
GoogLeNet+SVM [59]	2017		92.70±0.60	94.31±0.89
AlexNet [62]	2017			95.00±1.74
ResNet [62]	2017			97.19±0.57
Fusion by Addition [21]	2017			97.42±1.79
VGG-16+EMR [20]	2017			98.14
SPP-net+MKL [26]	2018			96.38
MCNN [63]	2018			96.66±0.90
OverfeatL + IFK [64]	2018			98.91
Triplet networks [14]	2018			97.99±0.53
VGG-16 + IFK [23]	2018			98.57±0.34
D-DSML-CaffeNet [24]	2018			95.76±1.70
VGG-16+MSCP [65]	2018			98.36±0.58
WSPM-CRC (ResNet152) [66]	2019			97.95
DDRL-AM (ResNet18) [67]	2019			99.05±0.08
ARCNet-VGG16 [40]	2019		96.81±0.14	99.12±0.40
Siamese ResNet50+RD [68]	2019		91.71	94.50
CTFCNN [69]	2019			98.44±0.58
CapsNet (Inception-v3) [70]	2019		97.59±0.16	99.05±0.24
Fine-tuning VGG16 [71]	2020		96.57±0.38	97.14±0.48
GBNet [71]	2020		95.71±0.19	96.90±0.23
GBNet + global feature [71]	2020		97.05±0.19	98.57±0.48
CAE ELM+CNN [41]	2019	90.18		
CAD+DenseNet [47]	2020		**98.57±0.33**	99.16±0.27
EfficientNetB3-Basic [ours]	2020	94.11±0.15	97.63±0.06	98.73±0.20
EfficientNetB3-Attn-2 [ours]	2020	**95.60±0.31**	97.90±0.36	**99.21±0.22**

TABLE 6. Comparison of EffecientNet-B3-Attn-2 model with the state-of-the-art performance on the KSA dataset.

Method	Year	Training ratio		
		20%	50%	80%
CS Multifeature Fusion [74]	2014		91.69	90.77
Pre-trained CNN+SAE [75]	2016		94.77	94.92
DAN with adaptation [57]	2017		94.36+0.41	95.33+0.79
Pretrained CNN+SVM [57]	2017		94.52	94.46
Multi-input SqueezeNet-Attn [72]	2019	92.15	93.48	
SqueezeNet-Attn [45]	2020	94.25	96.62	96.77
EfficientNetB3-Basic [ours]	2020	95.77±0.12	97.29±0.06	97.68±0.32
EfficientNetB3-Attn-2 [ours]	2020	**96.38±0.18**	**97.65±0.17**	**97.99±0.27**

it is a large dataset that contains 10,000 images, making it very difficult to train using the 80%-20% split using hardware available to us.

A comparative evaluation against several state-of-the-art classification methods on the UC Merced land-use dataset is shown in Table 5. We find that the proposed

TABLE 7. Comparison of EffecientNet-B3-Attn-2 model with the state-of-the-art performance on the RSSCN7 dataset.

Method	Year	Training ratio		
		20%	50%	80%
DBN based feature-selection [10]	2015		77.00	
CaffeNet [59]	2017	85.57±0.95	88.25±0.62	
VGG-VD-16 [1]	2017	83.98±0.87	87.18±0.94	
GoogLeNet [59]	2017	82.55±1.11	85.84±0.92	
TEX-Net-LF [76]	2018	92.45±0:45	94.0±0.57	
Resnet + Hybrid-KCRC (Hellinger) [73]	2018		**98.37**	
WSPM-CRC (ResNet152) [66]	2019		93.90	
Fine-tune MobileNet V2 [77]	2019	89.04±0.17	92.46±0.66	
SE-MDPMNet [77]	2019	92.65±0.13	94.71±0.15	
Dual Attention aware features [48]	2020	91.07±0.65	93.25±0.28	
Contourlet CNN [78]	2020		95.54±0.71	
EfficientNetB3-Basic [ours]	2020	92.06±0.39	94.39±0.10	95.18±0.21
EfficientNetB3-Attn-2 [ours]	2020	**93.30±0.19**	96.17±0.23	**96.89±0.20**

TABLE 8. Comparison of EffecientNet-B3-Attn-2 model with the state-of-the-art performance on the OPTIMAL-31 dataset.

Method	Year	Training ratio		
		20%	50%	80%
Fine-tuning GoogLeNet [61]	2015			82.57±0.12
VGG16 [59]	2017			89.12±0.35
Fine-tuning AlexNet [40]	2019			81.22±0.19
Fine-tuning GoogLeNet [40]	2019			82.57±0.12
Fine-tuning VGG16 [40]	2019			87.45±0.45
ARCNet- AlexNet [40]	2019			85.75+0.35
ARCNet- ResNet [40]	2019			91.28+0.45
ARCNet-VGG16 [40]	2019			92.70±0.35
Fine-tuning VGG16 [71]	2020			89.52±0.26
GBNet [71]	2020			91.40±0.27
GBNet + global feature [71]	2020			93.28±0.27
MAA-CNN [46]	2020			95.70±0.54
EfficientNetB3-Basic [ours]	2020	84.02±0.93	90.80±0.84	94.76±0.26
EfficientNetB3-Attn-2 [ours]	2020	**84.43±0.56**	**91.80±0.54**	**95.86±0.22**

EffecientNet-B3 CNN-Attn-2 achieves a 95.60±0.31 under the 20% training ratio and 97.90±0.36 under the 50% training ratio is competitive with most of the state-of-the-art methods. However, at an 80% training ratio, the accuracy of 99.21±0.22% is outperforming all other state-of-the-art methods.

Impressive results are also achieved for the KSA dataset. As can be seen in Table 6, for 20% and compared with Multi-input SqueezeNet-Attn [72] and SqueezeNet-Attn [45], the new proposed EfficientNet-B3-Attn-2 method outperformed them which prove that the attention mechanism is effective, and when it is combined with a

TABLE 9. Comparison of EffecientNet-B3-Attn-2 model with the state-of-the-art performance on the WHU-RS19 dataset.

Method	Year	Training ratio		
		40%	60%	80%
CaffeNet [59]	2017	95.11±1.20	96.24±0.56	
VGG-VD-16 [59]	2017	95.44±0.60	96.05±0.91	
GoogLeNet [59]	2017	93.12±0.82	94.71±1.33	
Fusion by Addition [21]	2017		98.65±0.43	
TEX-Net-LF [76]	2018	98.48±0:37	98.88±0:49	
Resnet + Hybrid-KCRC (Hellinger) [73]	2018		92.87	
WSPM-CRC (ResNet152) [66]	2019		98.32	
Fine-tune MobileNet V2 [77]	2019	96.82±0.35	98.14±0.33	
SE-MDPMNet [77]	2019	98.46±0.21	98.97±0.24	
Resnet101-FSL [79]	2019		98.77	
ARCNet-VGGNet16 [40]	2019		**99.75±0.20**	
GBNet + global feature [71]	2020		99.25±0.50	
EfficientNetB3-Basic [ours]	2020	97.28±0.24	97.68±0.10	98.95±0.20
EfficientNetB3-Attn-2 [ours]	2020	**98.60±0.40**	98.68±0.93	**99.47±0.20**

strong CNN network such as EffecientNet CNN yields an impressive improvement. As for 50% and 80% splits, the level of improvement is not as impressive, however, it still outperforms the state-of-the-art methods: pretrained CNN+SVM [57] and DAN with adaptation [57].

On the RSSCN7 dataset, as seen in Table 7, the proposed method obtains 96.89±0.20% for an 80% training ratio, which is better than the model without attention. As for the 20% training ratio, the achieved accuracy of 93.30±0.19% is outperforming the latest state-of-the-art.

However, for the 50% training ratio, we get lower performance than the Resnet+Hybrid-KCRC (Hellinger) method [73]. This can be explained by the fact that this method uses the images with their original resolution of 400 × 400, whereas we have resized the images to 256 × 256 (in fact we resized all datasets to 256 × 256 as is done by most other works).

The process of downsampling the original image from 400 × 400 to 256 × 256 pixels for the RSSCN7 dataset in our preprocessing step causes some loss of important information and has a negative impact on the classification result. Of course, it also minimizes the required training time. Besides, the RSSCN7 dataset is considered a challenging dataset mainly because it contains many classes such as (industry), (parking), and (resident) that are visually very similar. Thus, these classes usually share similar features, and the scenes from the industrial area are easy to be tangled with scenes from the residential area and the parking lots.

As for OPTIMAL-31 dataset, we can see from Table 8, that the proposed method outperforms all existing state-of-the-art methods and its achieves a 95.86±.22% accuracy under the 80% training ratio. OPTIMAL-31 dataset is one of most challenging dataset because of the high data variability in terms of different sensors, different scales, and so on. Thus these results demonstrate the effectiveness and power of our proposed model.

The results for the WHU-RS19 dataset are presented in Table 9. The EffecientNet-B3-Attn-2 model obtains 98.60±0.40% and 98.68±0.93% accuracy under the 40% and 60% training ratio, respectively, which is competitive with most state-of-the-art. In fact, our proposed EffecientNet-B3-Attn-2 misclassifies only one image under the 80% training ratio obtaining 99.47±0.20% classification accuracy, and only two images more than the ARCNet-VGGNet16 method [71] under the 60% training ratio.

TABLE 10. Comparison of EffecientNet-B3-Attn-2 model with the state-of-the-art performance on the AID dataset.

Method	Year	Training ratio	
		20%	50%
CaffeNet [59]	2017	86.86±0.47	89.53±0.31
VGG-VD-16 [59]	2017	86.59±0.29	89.64±0.36
GoogLeNet+SVM [59]	2017	83.44±0.40	86.39±0.55
Fusion by Addition [21]	2017		91.87±0.36
salM^{3}LBP-CLM [80]	2017	86.92±0.35	89.76±0.45
DCA(VGGNet) [21]	2017		91.86±0.28
TEX-Net-LF (ResNet) [76]	2018	93.81±0.12	95.73±0.16
RTN(VGG16) [81]	2018	92.44	
SAL-TS-Net (GoogleNet) [82]	2018	94.09±0.34	95.99±0.35
VGG-16+MSCP [65]	2018	91.52±0.21	94.42±0.17
MCNN [63]	2018		91.80±0.22
Multilevel Fusion [23]	2018		95.36±0.22
ARCNet-VGG16 [40]	2019	88.75±0.40	93.10±0.55
MRBF [83]	2019		87.26±0.42
FACNN [84]	2019		95.45±0.11
SF-CNN(VGGNet) [85]	2019	93.60±0.12	96.66±0.11
SCCov [86]	2019	93.12±0.25	96.10±0.16
RSFJR [87]	2019		**96.81±1.36**
ADFF [88]	2019	93.68±0.29	94.75±0.25
CTFCNN [69]	2019		94.91±0.24
CNN-CapsNet [70]	2019	93.79±0.13	96.32±0.12
WSPM-CRC (ResNet152) [66]	2019		95.11
Fine-tuning VGG16 [71]	2020	89.49±0.34	93.60±0.64
GBNet + global feature [71]	2020	92.20±0.23	95.48±0.12
Dual Attention-Aware Net [48]	2020	94.36±0.54	95.53±0.30
VGG-VD16 [49]	2020	94.75±0.23	96.93±0.16
MAA-CNN [46]	2020	95.54±0.08	97.48±0.07
CAD+DenseNet [47]	2020	95.73±0.22	97.16±0.26
EfficientNetB3-Basic [ours]	2020	93.43±0.33	95.37±0.41
EfficientNetB3-Attn-2 [ours]	2020	**94.45±0.76**	96.56±0.12

For this dataset, we discovered that there are different versions of the dataset. The version we are using has 19 classes with 50 images per class, i.e., a total of 950 images. Whereas some of the methods like ARCNet-VGGNet16 and [71] and GBNet + global feature [71] are using a version with 1005 images. The higher number of images partially explains the higher performance, as we know that deep learning models perform better when trained on more data. Unfortunately, the version with 1005 images is not available for download. Thus, we could not use it for a fair comparison.

Finally, for the AID dataset, which is relatively the largest RS scene dataset compared with other datasets, we use only 20%-80, and 50%-50% for train-test splits, as shown in Table 10. Our proposed method achieves 94.45+0.76 and 96.56+0.12 accuracy under these splits respectively, which is also better than most previous methods. However, three state-of-the-art methods have achieved better results than our method. These methods have appeared very recently and our method is not able to compete with them.

V. CONCLUSION

This paper proposes a novel deep learning model for the classification of RS scenes based on the EfficientNet CNN combined with an attention mechanism. We investigate two versions EfficientNet-B3-Attn-1 and EfficientNet-B3-Attn-2. In the EfficientNet-B3-Attn-1 model, the attention mechanism is added to the last feature map, whereas in the EfficientNet-B3-Attn-2, it is added at the end of layer 262. Thus EfficientNet-B3-Attn-2 has two branches; the main branch without attention and a secondary branch attached to the end of layer 262 that uses attention.

The results achieved with the EfficientNet-B3-Attn-2 model on six popular remote sensing datasets, namely UC Merced, KSA, OPTIMAL-31, and RSSCN7, have outperformed state-of-the-art. For the WHU-RS19 we have outperformed the methods that use the same dataset version that contains 950 image in total, but not the methods that use the dataset version containing 1005 images. Finally, for the AID dataset, we have also achieved better results than all previous methods, except three state-of-the-art methods that have appeared very recently.

Future developments include combining the proposed method with some of the novel techniques introduced in the very recent work CAD+DenseNet [47]. A second direction is adding data augmentation techniques as in MAA-CNN [46]. Finally, another possible improvement is inserting the attention mechanism in every MBConv block of the EfficientNet-B3 CNN model, similar to the inclusion of the Squeeze-and-Excitation branch in every block.

ACKNOWLEDGMENT

This work was supported by the Researchers Supporting Project through the King Saud University, Riyadh, Saudi Arabia under Grant RSP-2020/69.

REFERENCES

[1] A. Singh, "Review article digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, Jun. 1989, doi: 10.1080/01431168908903939.

[2] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017, doi: 10.1109/JPROC.2017.2675998.

[3] 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 J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3735–3756, 2020, doi: 10.1109/JSTARS.2020.3005403.

[4] J. Rogan and D. Chen, "Remote sensing technology for mapping and monitoring land-cover and land-use change," *Progr. Planning*, vol. 61, no. 4, pp. 301–325, May 2004, doi: 10.1016/S0305-9006(03)00066-7.

[5] H. Zhang, H.-J. Song, and B.-C. Yu, "Application of hyper spectral remote sensing for urban forestry monitoring in natural disaster zones," in *Proc. Int. Conf. Comput. Manage. (CAMAN)*, Wuhan, China, May 2011, pp. 1–4, doi: 10.1109/CAMAN.2011.5778867.

[6] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and POIs," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Beijing, China, 2012, p. 186, doi: 10.1145/2339530.2339561.

[7] Y. Wang, L. Zhang, X. Tong, L. Zhang, Z. Zhang, H. Liu, X. Xing, and P. T. Mathiopoulos, "A three-layered graph-based learning approach for remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6020–6034, Oct. 2016, doi: 10.1109/TGRS.2016.2579648.

[8] Y. Li, Y. Zhang, C. Tao, and H. Zhu, "Content-based high-resolution remote sensing image retrieval via unsupervised feature learning and collaborative affinity metric fusion," *Remote Sens.*, vol. 8, no. 9, p. 709, Aug. 2016, doi: 10.3390/rs8090709.

[9] F. P. S. Luus, B. P. Salmon, F. van den Bergh, and B. T. J. Maharaj, "Multiview deep learning for land-use classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2448–2452, Dec. 2015, doi: 10.1109/LGRS.2015.2483680.

[10] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2321–2325, Nov. 2015, doi: 10.1109/LGRS.2015.2475299.

[11] H. Wu, B. Liu, W. Su, W. Zhang, and J. Sun, "Deep filter banks for land-use scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1895–1899, Dec. 2016, doi: 10.1109/LGRS.2016.2616440.

[12] F. Zhang, B. Du, and L. Zhang, "Scene classification via a gradient boosting random convolutional network framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1793–1802, Mar. 2016, doi: 10.1109/TGRS.2015.2488681.

[13] Y. Zhong, F. Fei, and L. Zhang, "Large patch convolutional neural networks for the scene classification of high spatial resolution imagery," *J. Appl. Remote Sens.*, vol. 10, no. 2, Apr. 2016, Art. no. 025006, doi: 10.1117/1.JRS.10.025006.

[14] Y. Liu and C. Huang, "Scene classification via triplet networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 220–237, Jan. 2018, doi: 10.1109/JSTARS.2017.2761800.

[15] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, Nov. 2015, doi: 10.3390/rs71114680.

[16] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning Earth observation classification using ImageNet pretrained networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 105–109, Jan. 2016, doi: 10.1109/LGRS.2015.2499239.

[17] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," 2015, *arXiv:1508.00092*. [Online]. Available: http://arxiv.org/abs/1508.00092

[18] K. Nogueira, O. A. B. Penatti, and J. A. D. Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognit.*, vol. 61, pp. 539–556, Jan. 2017, doi: 10.1016/j.patcog.2016.07.001.

[19] G. J. Scott, M. R. England, W. A. Starms, R. A. Marcum, and C. H. Davis, "Training deep convolutional neural networks for land–cover classification of high-resolution imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 4, pp. 549–553, Apr. 2017, doi: 10.1109/LGRS.2017.2657778.

[20] G. Wang, B. Fan, S. Xiang, and C. Pan, "Aggregating rich hierarchical features for scene classification in remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 9, pp. 4104–4115, Sep. 2017, doi: 10.1109/JSTARS.2017.2705419.

[21] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for VHR remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4775–4784, Aug. 2017, doi: 10.1109/TGRS.2017.2700322.

[22] Q. Weng, Z. Mao, J. Lin, and X. Liao, "Land-use scene classification based on a CNN using a constrained extreme learning machine," *Int. J. Remote Sens.*, vol. 39, no. 19, pp. 6281–6299, Oct. 2018, doi: 10.1080/01431161.2018.1458346.

[23] Y. Yu and F. Liu, "Aerial scene classification via multilevel fusion based on deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 287–291, Feb. 2018, doi: 10.1109/LGRS.2017.2786241.

[24] Z. Gong, P. Zhong, Y. Yu, and W. Hu, "Diversity-promoting deep structural metric learning for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 371–390, Jan. 2018, doi: 10.1109/TGRS.2017.2748120.

[25] E. Li, J. Xia, P. Du, C. Lin, and A. Samat, "Integrating multilayer features of convolutional neural networks for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5653–5665, Oct. 2017, doi: 10.1109/TGRS.2017.2711275.

[26] Q. Liu, R. Hang, H. Song, and Z. Li, "Learning multiscale deep features for high-resolution satellite image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 117–126, Jan. 2018, doi: 10.1109/TGRS.2017.2743243.

[27] G. Cheng, Z. Li, X. Yao, K. Li, and Z. Wei, "Remote sensing image scene classification using bag of convolutional features," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1735–1739, Oct. 2017, doi: 10.1109/LGRS.2017.2731997.

[28] Y. Liu, Y. Liu, and L. Ding, "Scene classification based on two-stage deep feature fusion," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 183–186, Feb. 2018, doi: 10.1109/LGRS.2017.2779469.

[29] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018, doi: 10.1109/TGRS.2017.2783902.

[30] P.-W. Hsiao and C.-P. Chen, "Effective attention mechanism in dynamic models for speech emotion recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 2526–2530, doi: 10.1109/ICASSP.2018.8461431.

[31] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 10, 2020, doi: 10.1109/TNNLS.2020.3019893.

[32] H. Nam, J.-W. Ha, and J. Kim, "Dual attention networks for multimodal reasoning and matching," 2016, *arXiv:1611.00471*. [Online]. Available: http://arxiv.org/abs/1611.00471

[33] Y. Zhu, C. Zhao, H. Guo, J. Wang, X. Zhao, and H. Lu, "Attention CoupleNet: Fully convolutional attention coupling network for object detection," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 113–126, Jan. 2019, doi: 10.1109/TIP.2018.2865280.

[34] M. Pedersoli, T. Lucas, C. Schmid, and J. Verbeek, "Areas of attention for image captioning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1251–1259, doi: 10.1109/ICCV.2017.140.

[35] Y. Peng, X. He, and J. Zhao, "Object-part attention model for fine-grained image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1487–1500, Mar. 2018, doi: 10.1109/TIP.2017.2774041.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*. [Online]. Available: http://arxiv.org/abs/1706.03762

[37] G. Zhang, S. Lu, and W. Zhang, "CAD-Net: A context-aware detection network for objects in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10015–10024, Dec. 2019, doi: 10.1109/TGRS.2019.2930982.

[38] Y. Su, Y. Wu, M. Wang, F. Wang, and J. Cheng, "Semantic segmentation of high resolution remote sensing image based on batch-attention mechanism," in *Proc. IGARSS-IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 3856–3859, doi: 10.1109/IGARSS.2019.8898198.

[39] X. Qi, K. Li, P. Liu, X. Zhou, and M. Sun, "Deep attention and multi-scale networks for accurate remote sensing image segmentation," *IEEE Access*, vol. 8, pp. 146627–146639, 2020, doi: 10.1109/ACCESS.2020.3015587.

[40] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1155–1167, Feb. 2019, doi: 10.1109/TGRS.2018.2864987.

[41] H. Wang, Y. Miao, H. Wang, and B. Zhang, "Convolutional attention in ensemble with knowledge transferred for remote sensing image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 643–647, Apr. 2019, doi: 10.1109/LGRS.2018.2878350.

[42] C. Zhang, Q. Wang, and X. Li, "A multi-task architecture for remote sensing by joint scene classification and image quality assessment," in *Proc. IGARSS-IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 10055–10058, doi: 10.1109/IGARSS.2019.8898659.

[43] D. Guo, Y. Xia, and X. Luo, "Scene classification of remote sensing images based on saliency dual attention residual network," *IEEE Access*, vol. 8, pp. 6344–6357, 2020, doi: 10.1109/ACCESS.2019.2963769.

[44] H. Hu, Z. Li, L. Li, H. Yang, and H. Zhu, "Classification of very high-resolution remote sensing imagery using a fully convolutional network with global and local context information enhancements," *IEEE Access*, vol. 8, pp. 14606–14619, 2020, doi: 10.1109/ACCESS.2020.2964760.

[45] A. S. Alswayed, H. S. Alhichri, and Y. Bazi, "SqueezeNet with attention for remote sensing scene classification," in *Proc. 3rd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, Riyadh, Saudi Arabia, Mar. 2020, pp. 1–4, doi: 10.1109/ICCAIS48893.2020.9096876.

[46] F. Li, R. Feng, W. Han, and L. Wang, "An augmentation attention mechanism for high-spatial-resolution remote sensing image scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3862–3878, 2020, doi: 10.1109/JSTARS.2020.3006241.

[47] W. Tong, W. Chen, W. Han, X. Li, and L. Wang, "Channel-attention-based DenseNet network for remote sensing image scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4121–4132, 2020, doi: 10.1109/JSTARS.2020.3009352.

[48] Y. Gao, J. Shi, J. Li, and R. Wang, "Remote sensing scene classification with dual attention-aware network," in *Proc. IEEE 5th Int. Conf. Image, Vis. Comput. (ICIVC)*, Beijing, China, Jul. 2020, pp. 171–175, doi: 10.1109/ICIVC50857.2020.9177460.

[49] J. Ji, T. Zhang, L. Jiang, W. Zhong, and H. Xiong, "Combining multilevel features for remote sensing image scene classification with attention model," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 9, pp. 1647–1651, Sep. 2020, doi: 10.1109/LGRS.2019.2949253.

[50] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114. Accessed: Oct. 11, 2020. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[52] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[54] A. Kızrak. (Jan. 7, 2020). *Comparison of Activation Functions for Deep Neural Networks*. Accessed: Oct. 11, 2020. [Online]. Available: https://towardsdatascience.com/comparison-of-activation-functions-for-deep-neural-networks-706ac4284c8a

[55] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze- and-excitation networks," 2017, *arXiv:1709.01507*. [Online]. Available: http://arxiv.org/abs/1709.01507

[56] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (GIS)*, New York, NY, USA, 2010, pp. 270–279, doi: 10.1145/1869790.1869829.

[57] E. Othman, Y. Bazi, F. Melgani, H. Alhichri, N. Alajlan, and M. Zuair, "Domain adaptation network for cross-scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4441–4456, Aug. 2017, doi: 10.1109/TGRS.2017.2692281.

[58] H. Alhichri. *KSA Remote Sensing Datasets*. Alhichri Research Page. Accessed: Dec. 1, 2020. [Online]. Available: http://alhichri.36bit.com/ksa_dataset.html

[59] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, ''AID: A benchmark data set for performance evaluation of aerial scene classification,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017, doi: 10.1109/TGRS.2017.2685945.

[60] H. Alhichri. *Remote Sensing Datasets*. Alhichri Research Page. Accessed: Dec. 1, 2020. [Online]. Available: http://alhichri.36bit.com/research.html

[61] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, ''Land Use classification in remote sensing images by convolutional neural networks,'' *ArXiv*, vol. abs/1508.00092, p. 12, 2015.

[62] Y. Liang, S. T. Monteiro, and E. S. Saber, ''Transfer learning for high resolution aerial image classification,'' in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Washington, DC, USA, Oct. 2016, pp. 1–8, doi: 10.1109/AIPR.2016.8010600.

[63] Y. Liu, Y. Zhong, and Q. Qin, ''Scene classification based on multiscale convolutional neural network,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7109–7121, Dec. 2018, doi: 10.1109/TGRS.2018.2848473.

[64] Z. Yang, X.-D. Mu, and F.-A. Zhao, ''Scene classification of remote sensing image based on deep network and multi-scale features fusion,'' *Optik*, vol. 171, pp. 287–293, Oct. 2018, doi: 10.1016/j.ijleo.2018.06.024.

[65] N. He, L. Fang, S. Li, A. Plaza, and J. Plaza, ''Remote sensing scene classification using multilayer stacked covariance pooling,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 6899–6910, Dec. 2018, doi: 10.1109/TGRS.2018.2845668.

[66] B.-D. Liu, J. Meng, W.-Y. Xie, S. Shao, Y. Li, and Y. Wang, ''Weighted spatial pyramid matching collaborative representation for remote-sensing-image scene classification,'' *Remote Sens.*, vol. 11, no. 5, p. 518, Mar. 2019, doi: 10.3390/rs11050518.

[67] J. Li, D. Lin, Y. Wang, G. Xu, Y. Zhang, C. Ding, and Y. Zhou, ''Deep discriminative representation learning with attention map for scene classification,'' *Remote Sens.*, vol. 12, no. 9, p. 1366, Apr. 2020, doi: 10.3390/rs12091366.

[68] Y. Zhou, X. Liu, J. Zhao, D. Ma, R. Yao, B. Liu, and Y. Zheng, ''Remote sensing scene classification based on rotation-invariant feature learning and joint decision making,'' *EURASIP J. Image Video Process.*, vol. 2019, no. 1, pp. 1–11, Dec. 2019, doi: 10.1186/s13640-018-0398-z.

[69] H. Huang and K. Xu, ''Combing triple-part features of convolutional neural networks for scene classification in remote sensing,'' *Remote Sens.*, vol. 11, no. 14, p. 1687, Jul. 2019, doi: 10.3390/rs11141687.

[70] W. Zhang, P. Tang, and L. Zhao, ''Remote sensing image scene classification using CNN-CapsNet,'' *Remote Sens.*, vol. 11, no. 5, p. 494, Feb. 2019, doi: 10.3390/rs11050494.

[71] H. Sun, S. Li, X. Zheng, and X. Lu, ''Remote sensing scene classification by gated bidirectional network,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 82–96, Jan. 2020, doi: 10.1109/TGRS.2019.2931801.

[72] D. A. Ajjaji, M. A. Alsaeed, A. S. Alswayed, and H. S. Alhichri, ''Multi-instance neural network architecture for scene classification in remote sensing,'' in *Proc. Int. Conf. Comput. Inf. Sci. (ICCIS)*, Sakaka, Saudi Arabia, Apr. 2019, pp. 1–5, doi: 10.1109/ICCISci.2019.8716411.

[73] B.-D. Liu, W.-Y. Xie, J. Meng, Y. Li, and Y. Wang, ''Hybrid collaborative representation for remote-sensing image scene classification,'' *Remote Sens.*, vol. 10, no. 12, p. 1934, Dec. 2018, doi: 10.3390/rs10121934.

[74] L.-J. Zhao, P. Tang, and L.-Z. Huo, ''Land-use scene classification using a concentric circle-structured multiscale bag-of-visual-Words model,'' *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 12, pp. 4620–4631, Dec. 2014, doi: 10.1109/JSTARS.2014.2339842.

[75] E. Othman, Y. Bazi, N. Alajlan, H. Alhichri, and F. Melgani, ''Using convolutional features and a sparse autoencoder for land-use scene classification,'' *Int. J. Remote Sens.*, vol. 37, no. 10, pp. 1977–1995, 2016, doi: 10.1080/01431161.2016.1171928.

[76] R. M. Anwer, F. S. Khan, J. van de Weijer, M. Molinier, and J. Laaksonen, ''Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification,'' *ISPRS J. Photogramm. Remote Sens.*, vol. 138, pp. 74–85, Apr. 2018, doi: 10.1016/j.isprsjprs.2018.01.023.

[77] B. Zhang, Y. Zhang, and S. Wang, ''A lightweight and discriminative model for remote sensing scene classification with multidilation pooling module,'' *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 8, pp. 2636–2653, Aug. 2019, doi: 10.1109/JSTARS.2019.2919317.

[78] M. Liu, L. Jiao, X. Liu, L. Li, F. Liu, and S. Yang, ''C-CNN: Contourlet convolutional neural networks,'' *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 21, 2020, doi: 10.1109/TNNLS.2020.3007412.

[79] W. Huang, Q. Wang, and X. Li, ''Feature sparsity in convolutional neural networks for scene classification of remote sensing image,'' in *Proc. IGARSS-IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 3017–3020, doi: 10.1109/IGARSS.2019.8898875.

[80] X. Bian, C. Chen, L. Tian, and Q. Du, ''Fusing local and global features for high-resolution scene classification,'' *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 2889–2901, Jun. 2017, doi: 10.1109/JSTARS.2017.2683799.

[81] Z. Chen, S. Wang, X. Hou, and L. Shao, ''Recurrent transformer networks for remote sensing scene categorisation,'' in *Proc. Brit. Mach. Vis. Conf.*, Newcastle upon Tyne, U.K., Sep. 2018.

[82] Y. Yu and F. Liu, ''Dense connectivity based two-stream deep feature fusion framework for aerial scene classification,'' *Remote Sens.*, vol. 10, no. 7, p. 1158, Jul. 2018, doi: 10.3390/rs10071158.

[83] C. Wang, W. Lin, and P. Tang, ''Multiple resolution block feature for remote-sensing scene classification,'' *Int. J. Remote Sens.*, vol. 40, no. 18, pp. 6884–6904, Sep. 2019, doi: 10.1080/01431161.2019.1597302.

[84] X. Lu, H. Sun, and X. Zheng, ''A feature aggregation convolutional neural network for remote sensing scene classification,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7894–7906, Oct. 2019, doi: 10.1109/TGRS.2019.2917161.

[85] J. Xie, N. He, L. Fang, and A. Plaza, ''Scale-free convolutional neural network for remote sensing scene classification,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6916–6928, Sep. 2019, doi: 10.1109/TGRS.2019.2909695.

[86] N. He, L. Fang, S. Li, J. Plaza, and A. Plaza, ''Skip-connected covariance network for remote sensing scene classification,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1461–1474, May 2020, doi: 10.1109/TNNLS.2019.2920374.

[87] J. Fang, Y. Yuan, X. Lu, and Y. Feng, ''Robust space–frequency joint representation for remote sensing image scene classification,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7492–7502, Oct. 2019, doi: 10.1109/TGRS.2019.2913816.

[88] R. Zhu, L. Yan, N. Mo, and Y. Liu, ''Attention-based deep feature fusion for the scene classification of high-resolution remote sensing images,'' *Remote Sens.*, vol. 11, no. 17, p. 1996, Aug. 2019, doi: 10.3390/rs11171996.

HAIKEL ALHICHRI (Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in computer science from the University of Saskatchewan, Saskatoon, SK, Canada, and the Ph.D. degree in systems' design engineering from the University of Waterloo, Waterloo, ON, Canada.

He has more than ten years of academic and industrial experience in the computer engineering field in Canada, United Arab Emirates, and Saudi Arabia. He has worked briefly in a research capacity for Hypercore systems now bought by Sierra systems, Saskatoon, SK, Canada, and Intelligent Mechatronic Systems Inc., Waterloo, ON, Canada. From 2003 to 2007, he was as an Assistant Professor of Information Technology with the American University in Dubai, Dubai, United Arab Emirates. From 2007 to 2010, he worked with Sharesoft Solutions FZ LLC in Dubai, Dubai, United Arab Emirates, as a Product Manager and the Business Development Director. From 2010 to 2013, he has also acted as a Scientific Advisor to the Innovation Center, KSU. Since March 2010, he has been with the College of Computer and Information Sciences, King Saud University (KSU), Riyadh, Saudi Arabia, as a Faculty of Computer Engineering. He is a member of the Advanced Lab for Intelligent Systems Research. His research interests include remote sensing, pattern recognition, and machine intelligence.

ASMA S. ALSWAYED (Member, IEEE) received the B.S. (Hons.) degree in networking and telecommunication systems from Princess Nora bint Abdul Rahman University (PNU), Riyadh, Saudi Arabia, in 2014. She is currently pursuing the M.S. degree in computer engineering with King Saud University (KSU), Riyadh.

From 2014 to 2015, she was a Collaborator Research Assistant with the BioMedical Informatics Research National Center for Computer and Applied Math, King Abdulaziz City for Science and Technology. She also works as an Advisor for government entities with the National Center for Performance Measurements (Adaa). Her research interests include remote sensing, deep learning, and machine intelligence.

Ms. Alswayed's awards and honors include the Best Paper Award titled (Multi-Instance Neural Network Architecture for Scene Classification in Remote Sensing) from the 2019 International Conference on Computer and Information Sciences (ICCIS).

YAKOUB BAZI (Senior Member, IEEE) received the State Engineer and M.Sc. degrees in electronics from the University of Batna, Batna, Algeria, in 1994 and 2000, respectively, and the Ph.D. degree in information and communication technology from the University of Trento, Trento, Italy, in 2005.

From 2000 to 2002, he was a Lecturer with the University of M'sila, M'sila, Algeria. From January to June 2006, he was a Postdoctoral Researcher with the University of Trento. From August 2006 to September 2009, he was an Assistant Professor with the College of Engineering, Al-Jouf University, Al-Jouf, Saudi Arabia. He is currently a Professor with the Computer Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. His research interests include machine learning and pattern recognition methods for signal/image processing and analysis.

Dr. Bazi is a Referee for several international journals.

NASSIM AMMOUR (Member, IEEE) received the B.Sc. and M.Sc. degrees in engineering from the University of Saad Dahleb, Blida, Algeria, in 1988 and 1995, respectively, and the Ph.D. degree from the National Polytechnic School of Algiers, Algeria, in 2009.

He has more than 20 years of teaching experience at the Electrical Engineering Department, Blida University, Algeria. In 2009, he joined the ALISR Laboratory, College of Computer and Information Sciences, KSU. His research interests include robotics, intelligent control, and autonomous industrial operations.

NAIF A. ALAJLAN (Senior Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in electrical engineering from King Saud University, Riyadh, Saudi Arabia, in 1998 and 2003, respectively, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2006. From 1998 to 2000, he was a Systems and Control Engineer with Saudi Basic Industries Company, Riyadh. Since 2000, he has been with King Saud University, where he was a Lecturer with the Electrical Engineering Department, from 2000 to 2003, was an Assistant Professor with the Electrical Engineering Department, and then, with the Computer Engineering Department, from 2007 to 2011, and has been an Associate Professor with the Computer Engineering Department, since 2011. He is the Founder and the Director of the Innovation Center and the Advanced Laboratory for Intelligent Systems Research, King Saud University. He is also a Founding Member of the Global Venture Laboratory Network, University of California at Berkeley.