Weather classification using meta-based Random forest fusion of transfer learning models

Rasha Talib Gdeeb [a,1,*]

[a] Department of Environmental Engineering, College of Engineering, University of Baghdad, Iraq
[1] rasha.talib@coeng.uobaghdad.edu.iq
* corresponding author

ARTICLE INFO

Article history
Received August 17, 202
Revised March 16,, 2024
Accepted March 23, 2024
Available online May 31, 2024

Keywords
Deep learning
Image processing
Transfer learning
Meta-based fusion
Weather classification

ABSTRACT

Weather classification into multiple categories is an essential task for many applications, including farming, military, transport, airlines, navigation, agriculture, etc. A few pieces of research give attention to this field and the current state-of-art methods have limitations, including low accuracy and limited weather conditions. In this study, a new weather classification meta-based fusion of the transfer deep learning model is introduced. The study takes into account all possible weather conditions and utilizes the fusion technique to improve the performance. First, the weather images are pre-processed and a data augmentation process is performed. These images are fed into five transfer deep learning models (XceptionNet, VGG16, ResNet50V2, InceptionV3, and DenseNet201). Then, the meta-based random forest fusion, the meta-based bagging fusion, and the score-level fusion are applied. Finally, all individual and fusion models are evaluated. Experiments were conducted on the WEAPD dataset which includes 11 categories. Results prove that the best performance is related to the meta-based ransom forest fusion method with 96% accuracy. The current study is also compared with the current state-of-art methods, and the comparison proves the robustness and high performance of the current study especially the fact that the current study achieves the best performance on the WEAPD dataset compared to studies worked on the same dataset. The current study proves that meta-based RF fusion is a promising methodology to address the weather classification problem. This outcome can be used by future study to improve the weather classification fusion and ensemble methodologies.

1. Introduction

Environment changes monitoring and detection is one of the most essential topics for climatologists especially in the last decade [1]–[3] . Different weather changes and conditions affect many lifestyles, including farming, transport, agriculture, tourism, outdoor activities, etc. Weather image classification, as a specific branch of environment change detection, is the process of classifying weather images into many categories (sunny, rainy, cloudy, snowy, etc.) [4]–[6]. Weather image classification is essential in many applications; farmers can decide about planting and harvest times based on weather classification results [7]. It also assists airlines to decide about flight's routes, and also helps tourists to plan their trip schedule [8].Computer science algorithms can help to design models for the aim of weather classification based on weather images [9], [10]. Image processing is the branch of computer science that can be utilized to make a good weather classification system. However, traditional image processing and machine learning (ML) techniques lack accuracy [11], can't handle huge data size, and may produce undesired responses if the training images have challenges (illumination variations, pose variations, scale

variations, etc.) [12]. However, deep learning (DL), as the new branch of machine learning and its visual deep models can be used for the aim of weather classification registering higher accuracy and lower error rates [13], [14]. Earlier methods of image-based weather classification relied on the traditional image processing techniques like edge detection, color-based processing, texture analysis, machine learning classifiers [15].

2. Related Works

Using computer science algorithms, including image processing, machine learning and deep learning in the field of climate change detection and specifically weather detection has received good attention from researchers. However, this field still needs more investigations and a deeper analysis since data (images) are increasing day after day. Wang et al. [16] proposed a multi-task learning methodology for weather classification mission. They collected a multi-class weather dataset of nine different climates. They utilized DenseNet and ResNet along with probability discrimination on each model's output to improve performance. They achieved an accuracy of 68.25%, 72.25% and 72.75 using ResNet50, ResNet101 and DenseNet121 models, respectively. However, not all weather conditions were taken into account in their research besides the low classification accuracy. Galeb et al. [17] applied a traditional convolutional neural network with simple ML models, including decision trees (DT) and support vector machines (SVM). They utilized a Kaggle dataset of only 1500 images and five categories with a split of 80% as a training set and 20% for test set. Experiments were conducted using different scenarios (Individual CNN, CNN+DT, CNN+SVM) and achieved an accuracy of 92%, 93%, and 94%, respectively. However, their dataset size and number of categories were small. A new dataset called "Weather phenomenon database (WEAPD)", consisting of 6877 images and 11 different categories was collected by Zhang et al. [14] in 2021 for weather classification purposes. The dataset included many challenges like complex background and different image variations. They also proposed a CNN-based model named MeteCNN consisting of traditional convolutional layers, max-pooling layer, batch normalizations, and global average pooling layers. However, although the simple architecture of their proposed model, they got an accuracy of 92%, and also they got 93% for precision, recall, and F1-score. Kalkan et al. [18] introduced a weather classification deep learning model to classify images into cloudy or clear (only two classes). They used a dataset of cloudy and not cloudy images taken from ground. Researchers utilized transfer learning models, including VGG16, MobileNet, ResNet152, and DenseNet-201, but the best performance was corresponding to VGG16 with 91.4%. No fusion or ensemble methods were used in their research. A weather forecasting review study was proposed by Jaseena and Kovoor [8]. They classified studies according to their methodology to statistical-based, artificial intelligence-based methods, and hybrid methods. They found that the deep learning-based methods achieved the best performance against other methods. Çetiner and Metlek [19] used the WEAPD dataset for weather classification based on deep learning and transfer learning. They classified weather into 11 categories and utilized the ResNet152V2 and achieved an accuracy of 88%. They didn't utilize any enhancements to improve performance. Later, Mashudi et al. [20] introduced a deep learning-based approach for weather classification using four deep models, including InceptionResNetV2, XceptionNet, MobileNet, and DenseNet201. They utilized the same WEAPD dataset as Zhang et al. [14], They achieved an accuracy of 83% as the best performance using DensNet201 model. However, they didn't apply any fusion or ensemble methods to enhance performance. Dalal et al. [7] proposed a modified deep learning method using the YOLO model and hyperparameters tuning along with the "Without-forgetting (LwF)" method. They used a small dataset of only 1499 images of only 5 categories with split them into 70%, 20%, and 10% as a training set, a validation set, and a test set, respectively. They got an accuracy of 99.19% but using a very small dataset. Reviewing these previous studies revealed several areas of improvement to enhance their performance. Some studies used a small dataset, others used a small number of classes (weather categories), while some studies used simple or traditional methods and stocked with low performance.

The current study contribution can be concluded In the current study, we suggest using a meta-based fusion method of the best transfer deep learning models in order to improve the performance of the current state-of-art. This study will also use the WEAPD dataset with 11 different weather

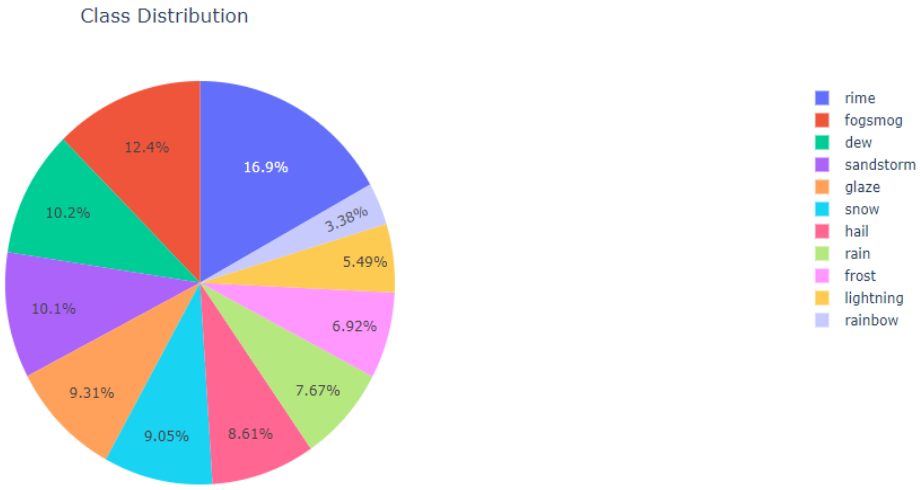
conditions which is the best choice of all used datasets in the literature. The rest of the paper will be organized as follows. First, the materials and methods will be listed and discussed; second, the experiments with corresponding results will also be presented and discussed; and finally, the study will be compared with the state-of-art methods and the conclusion will be derived.

3. Methods

3.1. Dataset

In this study, the WEAPD dataset [21] is used. This dataset consists of 6877 images distributed over 11 categories (Frost, Glaze, Snow, Rime, Rainbow, Rain, Dew, Fogsmog, Hail, Sandstorm, and Lightening). Fig. 1 shows some examples of different categories of the WEAPD dataset.

A.



B.

Fig. 1. WEAPD dataset A.Examples of categories of the WEAPD dataset, B. Class distribution [21]

3.2. Proposed Methodology

The current study utilizes five of the most accurate transfer deep learning models. These models include XceptionNet, VGG16, ResNet50V2, InceptionV3, and DenseNet201. The proposed methodology contains the following steps. First, the dataset images are resized into a specific size of 256*256, then they are preprocessed using the data augmentation method, including random scaling, random rotation and random flipping. Then, the transfer learning methodology is applied using the five proposed models which were already trained on the ImageNet dataset. The output of the transfer learning models is the feature vector that is introduced to the classification part. The classification part consists of dense layers (fully-connected layers) and softmax activation function. In the next step, the training process will be conducted for all models. After that, the score-level fusion is applied to the trained models in order to compute the fused score by combining the individual scores of all models. The meta-based fusion is next computed once using the RF algorithm and once using the bagging

method. In the last step, all models, including the individual and fusion models are evaluated using the performance evaluation metrics. Fig. 2 illustrates the proposed methodology.

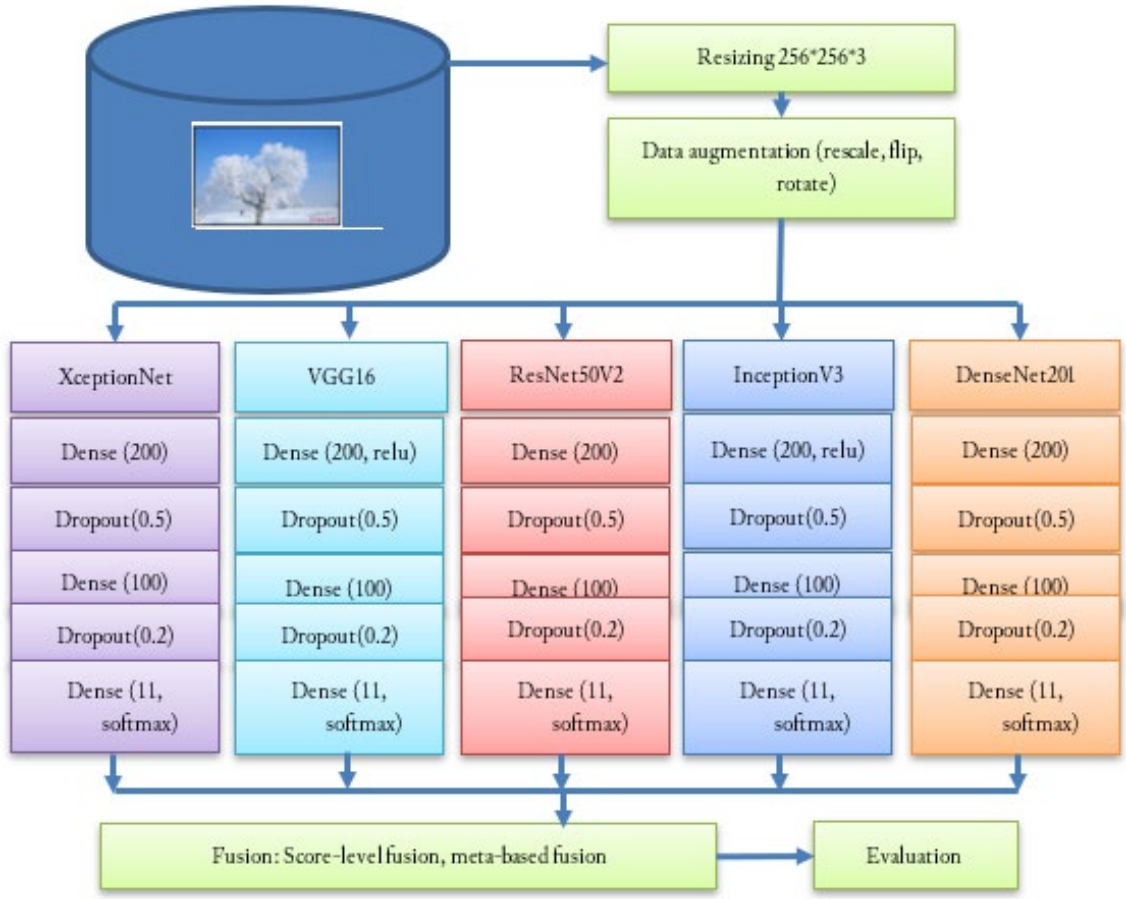
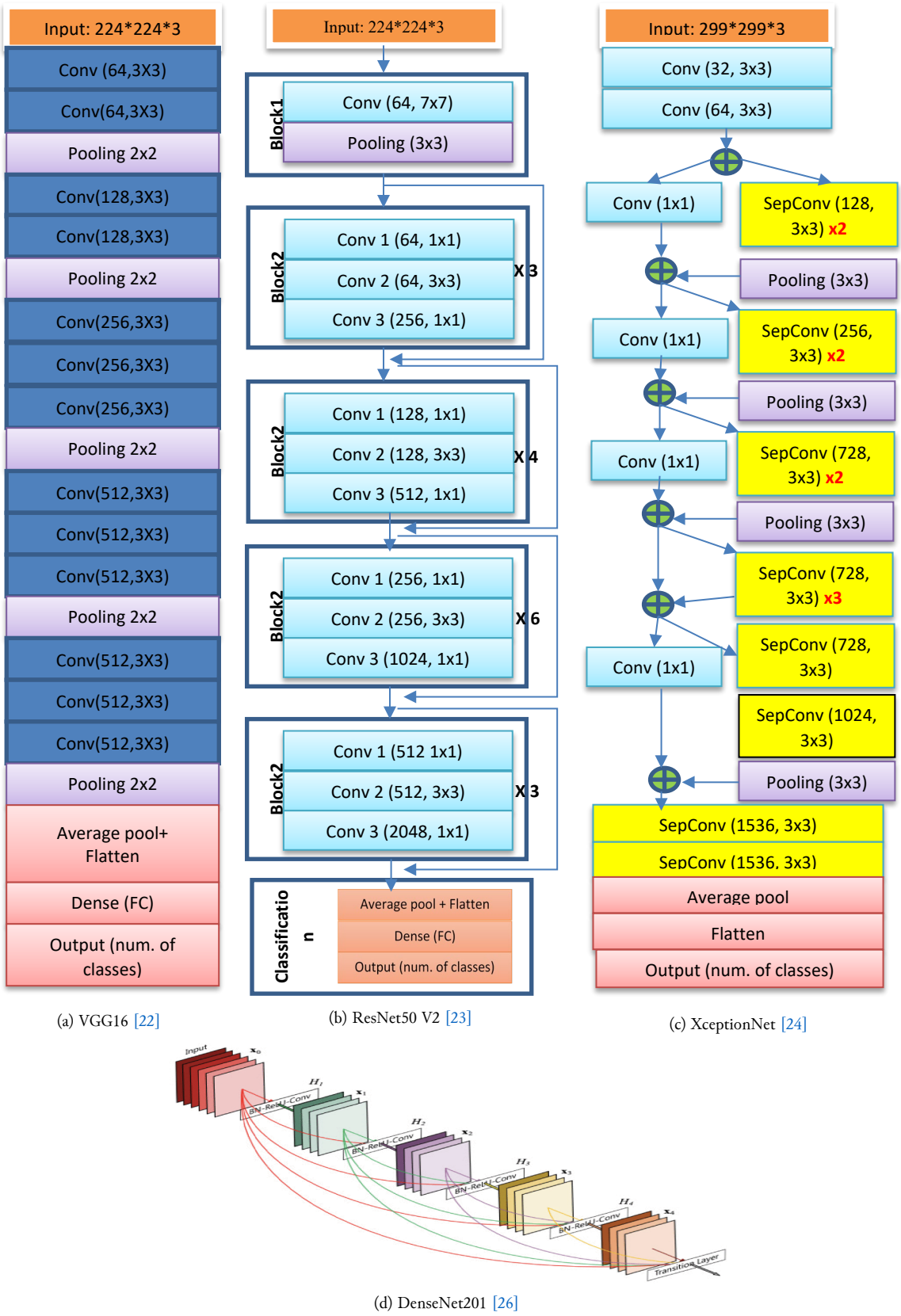


Fig. 2. The proposed methodology

3.3. Transfer Learning

Transfer learning is a DL technique that includes using pre-trained deep learning models (that are trained in a specific domain) in another domain using the knowledge (training weights) that is obtained from the first domain. In the current study, the pre-trained models that are trained on the ImageNet dataset will be reused in the weather classification domain (the current study's problem). VGG16 (Visual Geometry Group) model [22] is one of the most used transfer deep models, which is mainly a convolutional neural network model. VGG16 consists of 16 convolutional layers and 5 max pooling layers. The input layer is of size 224*224*3. The convolutional layers have filters of small sizes (3*3) and a stride of 1, while the max pooling layer has a stride of 2*2. Fig. 3(a) illustrates the architecture of the VGG16 model. ResNet50V2 [23] is another CNN-based model, consisting of 50 convolutional layers with identity connections (residual connections) that skip three convolutional layers to avoid the problem of vanishing gradient. Each convolutional layer has a batch normalization and Relu activation function. The first convolution has 64 filters with a size of 7*7 (which is small) in order to minimize the number of learnable parameters. The following stages have convolution blocks with [64, 64, 256], [128, 128, 512], [256, 256, 1024], and [512, 512, 2048] filters. Each stage begins with a convolution block where the first layer has strides=2, all others have strides=1. The network ends with a global average pooling, a 1000-way fully connected layer, and softmax. Fig. 3(b) illustrates the ResNet50V2 architecture. XceptionNet [24] has a total of 72 layers illustrated in Fig. 3(c). The input size of this model is 299*299*3 which is different from ResNets and VGG16. It starts with standard convolutions, followed by 8 inception modules with depthwise separable convolutions (with different numbers of filters).



(a) VGG16 [22] (b) ResNet50 V2 [23] (c) XceptionNet [24]

(d) DenseNet201 [26]

Fig. 3. DL models architecture

Separable convolutions are a type of convolutional layer factorizing the original convolution process into two separate operations: depthwise convolutions and pointwise convolutions. In the former, a single

filter is applied to each input separately, while in the second, a 1x1 convolution is applied to the output of the depthwise part (which is called the intermediate feature representation), and produces the final feature representation. After the feature extraction part, a global average pooling and fully connected layers exist. The number of filters in the modules increases stage by stage: it's 128, 256, 728, and 1024 in the last three stages. InceptionV3 [25] (Fig. 3(d)) is another convolutional network consisting of 48 layers with an initial 3x3 convolutional layers. The input of this model is of size 299*299*3. After initial convolutional layers, there are three Inception-A blocks, which include convolutions with filter sizes of 1x1, 3x3, and 5x5, and a pooling operation. After that, there are five Inception-B blocks, which are similar but have seven 1x1 convolutions instead of the 5x5 convolution, followed by two Inception-C blocks, which are similar to the Inception-A blocks but include a 1x3 and a 3x1 convolution instead of the 5x5 convolution. This model ends with average pooling, dropout, and a fully connected layer. In Fig. 3(d), the DenseNet201 model is illustrated [26]. It consists of 201 layers and dense blocks in which each layer is connected to every other layer, and transition layers by which the number and dimensions of the feature maps are reduced. Each dense block includes many convolution layers. Each convolution layer consists of batch normalization, ReLU activation function, a 1x1 convolution, second batch normalization, second ReLU activation, and a 3x3 convolution. The number of convolution layers in the dense blocks is 6, 12, 48, and 32. Each transition layer includes batch normalization, a 1x1 convolution, and a 2x2 average pooling. The model ends with global average pooling and a fully connected layer

3.4. Validation Method

Out of the training set of the dataset, a 20% split, containing 1097 images is used. The validation set is used during the training process after each training epoch to evaluate the performance of the model on a data which is not used in the training process. This operation is essential to ensure that the model is correctly trained and no overfitting is occurring. The validation images are of size 256*256 and have been shuffled each epoch and the performance of the trained model is monitored and assessed. The training and validation curves are also plotted in order to judge the individual trained models and choose the best trained models.

3.5. Meta-Based Fusion Methodology

The proposed meta-based RF fusion algorithm uses the scores of all individual models, then converts these scores into predictions using the "argmax" operation. After that, the predictions are stacked into a matrix, and the meta-model is fitted using the RF classifier. Again, the test predictions are stacked into a matrix and the meta-model is used to predict on the test set to get the final fused prediction as illustrated in Algorithm Meta-based RF fusion (Fig. 4).

Algorithm: Meta-based RF fusion algorithm

Inputs: xception_scores, vgg16_scores, ResNet_scores, Incp_scores, Dens_scores, test_gen.classes

Outputs: final_predictions

1. Convert scores to class predictions: xception_preds = ARGMAX(xception_scores, axis=1), vgg16_preds = ARGMAX(vgg16_scores, axis=1), ResNet_preds = ARGMAX(ResNet_scores, axis=1), Incp_preds = ARGMAX(Incp_scores, axis=1), Dens_preds = ARGMAX(Dens_scores, axis=1)
2. Stack predictions into a matrix: stacked_predictions = STACK_COLUMNS (xception_preds, vgg16_preds, ResNet_preds, Incp_preds, ens_preds)
3. Fit a meta-model using the random forest classifier: meta_model = RandomForestClassifier FIT(meta_model, stacked_predictions, test_gen.classes)
4. Stack test predictions into a matrix: stacked_test_predictions=STACK_COLUMNS (xception_preds, vgg16_preds,ResNet_preds, Incp_preds, Dens_preds)
5. Use meta-model to predict on test set: final_predictions = PREDICT(meta_model, stacked_test_predictions)

RETURN final_predictions

Fig. 4. Meta-based RF fusion algorithm.

For comparison purposes, we utilized two other different fusion methodologies, which are the score-level fusion and meta-based bagging fusion. In bagging fusion, the individual scores of the individual models are computed and then these scores are converted to class predictions using the argmax function. The class predictions are then stacked into one matrix and forwarded to a bagging classifier. The bagging classifier is then trained and then evaluated by mapping the predictions to the appropriate labels. For the score-level fusion, the individual scores of the individual models are first obtained and then weighted using a specific weight for each model (the weight is assigned based on the performance of each model so the model with higher performance gets a higher weight). Then, the final prediction is obtained using the argmax function and the mapping step is finally obtained

3.6. Performance Evaluation Metrics

Performance metrics play an important role in evaluating the trained models and judging their performance [27], [28]. These parameters help us to make a comprehensive overview of the designed models and let's know the cause of low performance.

Precision is a metric that measures the percentage of correctly classified positive samples out of all predicted positive samples. It is calculated as TP/(TP+FP). Recall, on the other hand, measures the percentage of correctly classified positive samples out of all actual positive samples. It is calculated as TP/(TP+FN). F1-score is a performance metric that combines Precision and Recall into a single metric and is calculated as 2*precision*recall / (precision + recall). Accuracy is the ratio of correctly classified samples to the total number of samples, regardless of the class. TP: is the number of true positives, TN: is the number of true negatives, FP: is the number of false positives, and FN is the number of false negatives.

In terms of true negatives, the specificity metric is calculated as the ratio of true negatives to the sum of the true negatives and false positives [29]. The final performance evaluation method is the confusion matrix which presents a breakdown of the results for each category in the classification problem, including precision, recall, and F1-score [30]–[33]. In a performance evaluation process, a low precision rate indicates a large number of false positives, while a low recall indicates a large number of false negatives. Low specificity means that the model is incorrectly identifying negative instances of the datasets [34].

4. Results and Discussion

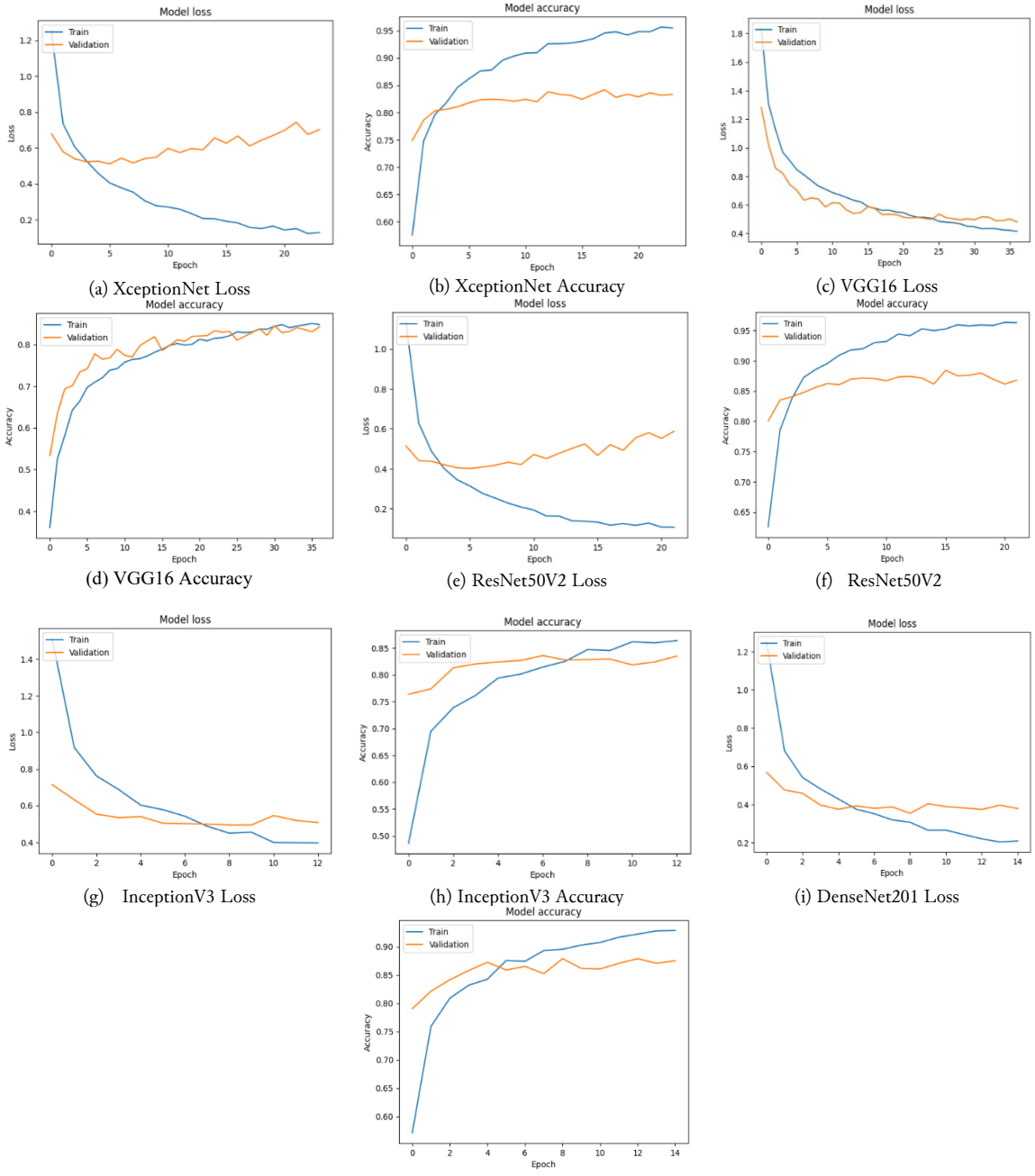
4.1. Training Parameters And Training Scenarios

The dataset is split into 80% as a training set, and 20% for a test set. A subset of 20% of the training set is used for the test. The data augmentation tasks are performed in order to increase the number of training images and make some variations in the training data and enhance the training process. The data augmentation process includes: rescaling to scope 0-1 by dividing all image values by 255, rotation with a range of 150, and a horizontal flip. Dataset training images are also shuffled in order to enhance the training process. The input size is unified for all images and models as 256*256, a batch size of 32 is also used for all models, the class mode is the "categorical" mode, and the output size is 11 neurons since the number of classes is 11. Besides that, the "Softmax" activation function is used in the last layer. All trained parameters of the transfer learning models are frozen. The activation function of the dense layers is "Relu". The training process includes using the following parameters: The used optimizer is "Adam", the loss function is the categorical cross-entropy, the training metrics is the "accuracy", and the training process includes a stop condition by which the training process will be stopped if the validation accuracy has a same or lower value than previous iterations for 6 epochs (patience = 6), the training process saves the best training results only, the monitor of the training process is the validation accuracy, and the number of epochs is 100 (but none of the models reached this number because of the early stop condition). According to these parameters, many training scenarios are performed, including training the individual models (XceptionNet, VGG16, ResNet50V2, InceptionV3, and DenseNet201) with the training set and validating it using the validation set. The test set will also be used to evaluate the

performance of the trained models. moreover, the score-level fusion, the meta-based RF fusion, and the meta-based Bagging fusion will also be performed and evaluated.

4.2. Results Of Performance Evaluation Of Individual Models

The early stop condition in the training process causes different convergence of the individual models. Fig. 5 shows the training accuracy and loss of the five trained models. XceptionNet model trained for 24 epochs, VGG16 trained for 28 epochs, ResNet50V2 needed only 22 epochs, InceptionV3 spent the least training time with only 13 epochs, while the DenseNet201 training epochs was 15. The training accuracies of all models are: 85.7%, 85.04%, 96.31%, 86.38%, and 92.85% for XceptionNet, VGG16, Resnet50V2, InceptionV3, and DenseNet201, respectively. ResNet50V2 registered the best training accuracy and the best validation accuracy with 88.42%. The second-best validation accuracy corresponds with the DenseNet201 with 87.88%. XceptionNet, VGG16, and InceptionV3 registered validation accuracies of 84.14%, 84.5%, and 83.59%, respectively.



(a) XceptionNet Loss (b) XceptionNet Accuracy (c) VGG16 Loss

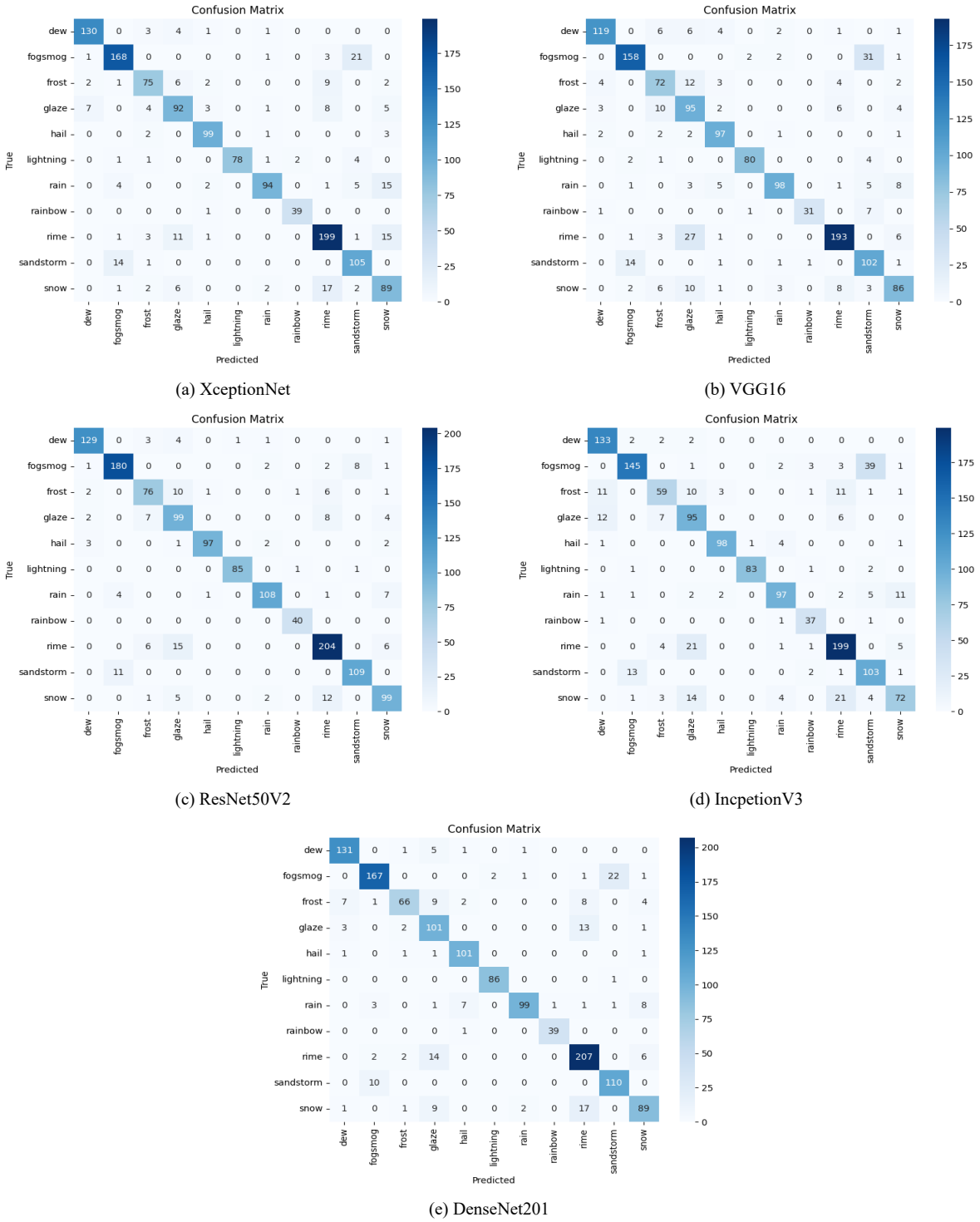
(d) VGG16 Accuracy (e) ResNet50V2 Loss (f) ResNet50V2

(g) InceptionV3 Loss (h) InceptionV3 Accuracy (i) DenseNet201 Loss

(j) DenseNet201

Fig. 5. Accuracy and loss curves of all trained transfer models

The confusion matrixes of evaluating trained models using test sets are shown in Fig. 6. The best results are related to the ResNet50V2 model with the least number of false positives and false negative errors.



(a) XceptionNet

(b) VGG16

(c) ResNet50V2

(d) IncpetionV3

(e) DenseNet201

Fig. 6. Confusion matrix of evaluating the trained models using the test set

The precision, recall, and F1-score metrics for all categories of the five trained models are illustrated in Table 1.

Table 1. Precision, Recall, and F1-score of all different categories of the trained models

	XceptionNet				VGG16				ResNet50V2				InceptionV3				DenseNet201			
	P%	*R%*	*F%*	*S%*	*P%*	*R%*	*F%*	*S%*	*P%*	*R%*	*F%*	*S%*	*P%*	*R%*	*F%*	*S%*	*P%*	*R%*	*F%*	*S%*
dew	93	94	93	99	92	86	89	99	94	93	93	99	84	96	89	98	96	94	95	99
Fogs-mog	88	87	87	98	89	81	85	98	92	93	93	99	89	75	81	98	94	94	94	99
frost	82	77	80	99	72	74	73	98	82	78	80	99	79	61	69	99	88	81	84	99
glaze	77	77	77	98	61	79	69	95	74	82	78	97	66	79	72	96	77	88	82	97
hail	91	94	93	99	85	92	89	99	98	92	95	100	95	93	94	100	98	95	97	99
Light-ning	100	90	95	100	96	92	94	100	99	98	98	100	99	95	97	100	100	98	99	100
rain	93	78	85	99	92	81	86	99	94	89	92	98	89	80	84	99	96	91	94	100
Rain-bow	95	97	96	100	97	78	86	100	95	100	98	100	82	86	84	99	98	100	99	100
rime	84	86	85	97	91	84	87	98	88	88	88	98	66	86	75	96	89	90	90	96
Sand-storm	76	88	81	97	67	85	75	96	92	91	92	99	66	86	75	96	93	93	93	98
snow	69	75	72	97	78	72	75	98	82	83	82	98	78	61	68	98	85	83	84	98
M- A	86	86	86	98	84	82	83	98	90	90	90	99	83	82	82	98	89	88	88	99
W-A	85	85	85	99	84	82	83	98	90	89	89	99	82	82	82	98	88	87	87	98

[a] M-A: macro average, W-A: Weighted average. P: precision, R: recall, F: F1-score, S: Specificity

Results of Table 1 prove that the model ResNet50V2 achieves the best performance in the case of all metrics (precision, recall, F1-score, and specificity). The mean average (M-AVG) precision, recall, and F1-score of the ResNet50V2 model is 90%, while the specificity of it is almost 99%. The worst model is the InceptionV3 model with only 82% for Precision, Recall, and F1-score, respectively. However, all individual models couldn't achieve a high performance (under different training options) which lead to going through the fusion models. We also computed the AUC for all models and got values of 99% to 100% with superiority of the ResNet50V2 model.

4.3. Results Of The Fusion Models

The confusion matrixes of the three fusion models are illustrated in Fig. 7. The best results are corresponding to the meta-based RF fusion model with the least number of false positives and false negatives errors. The results of the three different fusion methodologies are shown in Table 2.

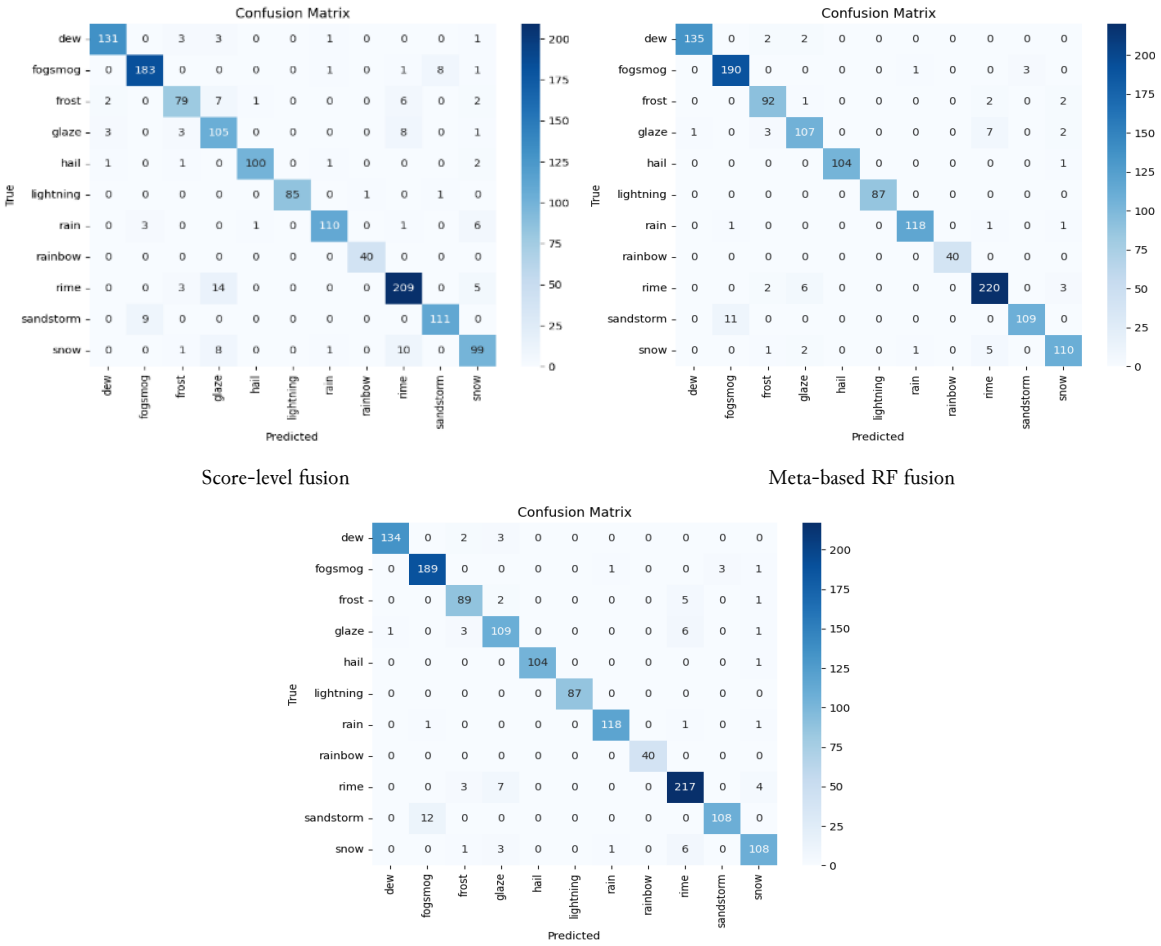


Fig. 7. Confusion matrix of the fusion models

Table 2. Precision, recall, and F1-score of all different categories of the trained models

	Score-level fusion				Meta-based fusion (RF)				Meta-based fusion (Bagging)			
	P%	*R%*	*F%*	*S%*	*P%*	*R%*	*F%*	*S%*	*P%*	*R%*	*F%*	*S%*
dew	96	94	95	99	99	97	98	100	99	96	98	100
Fogsmog	94	94	94	99	94	98	96	99	94	97	95	99
frost	88	81	84	99	92	95	93	99	91	92	91	99
glaze	77	88	82	97	91	89	90	99	88	91	89	99
hail	98	95	97	100	100	99	100	100	100	99	100	100
Lightning	100	98	99	100	100	100	100	100	100	100	100	100
rain	96	91	94	100	98	98	98	100	98	98	98	99
Rainbow	96	91	94	100	100	100	100	100	100	100	100	100
rime	89	90	90	98	94	95	94	99	92	94	93	98
Sandstorm	93	93	93	99	97	91	94	100	97	90	94	100
snow	85	83	84	99	92	92	92	100	92	91	92	100
M-AVG	92	92	92	99	96	96	96	100	96	95	95	99
W-AVG	91	91	91	100	96	96	96	100	95	95	95	99

The Meta-based fusion using the RF method has the best performance with 96% for precision, recall, and F1-score, while the meta-based bagging fusion method achieves closed results with 96% as precision, and 95% for recall and F1-score. However, each of the precision, recall, and F1-scores of the score-level fusion has a value of 92%. The meta-based RF fusion model enhanced the performance by 6% for all metrics (precision, recall, and F1-score) when compared to the best individual model (ResNet50V2). In terms of specificity, all fused models achieved a high score with a 100% score for the Meta-based fusion (RF) case.

In terms of test accuracy, the meta-based RF fusion achieves the best test accuracy of 96% as shown in Fig. 8. The test accuracy of the meta-based RF model enhanced the accuracy by 7% when compared to the best individual model (ResNet50V2).

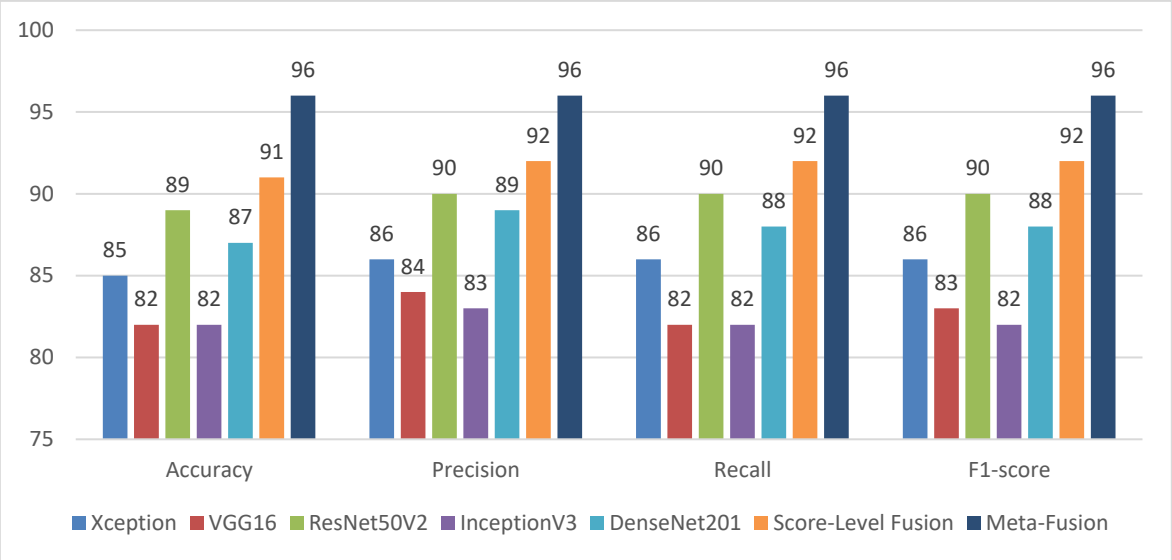


Fig. 8. Performance evaluation of all trained models (individual and fusion)

4.4. Discussion of the Individual Models Results

The confusion matrixes of Fig. 2 illustrate the following: In case of XceptionNet, the categories ("rime", "Snow", "rain", "glaze", "Sandstorm", "fogsmog", "frost") have more than 10 false negatives. The category with the highest number of FN errors is "fogsmog" in which 21 samples were classified as "sandstorm". However, the best category is "rainbow" with one false negative error. In terms of FP errors, the "sandstorm" category has the highest number of FP (33 errors). For VGG16, category "fogsmog" has 36 FN errors and 20 FP errors. "rims" category has the highest FN rate with 38 errors. Again, the "sandstorm" category has the highest FPs with 50 errors, but the least FNs with only 4 errors. ResNet50V2, which is the best model, contains 27 FN errors for "rime" category which is the category with the highest number of errors. On the other hand, "rainbow" category has no FN errors. "daze" category has the highest FP errors (35 errors), while other categories like "lightening", "rainbow", and "hail" have a small FP error rate. For InceptionV3 model, "fogsmog" has the highest FN error rate with 49 errors, while the category with the highest FP error rate is "sandstorm" with 52 errors. For DenseNet201 model, two classes achieve the highest FN error rate which are "fogsmog" and "snow" with 27 and 30 errors, respectively. Similarly, the "daze", and "rime" register high FP errors with 39, and 40, respectively. The general note about all models is that the similarities between some categories lead to some FP and FN errors. For example, the categories "fogsmog" and "sandstorm" caused most of model's errors due to their similarity in images since fog and sandstorm has very closed attitude in images. Other categories that caused errors due to similarities are: "snow" and "rime" (color white is the common texture), "rime" and "glaze" (too similar weather types and in some cases, even human get confused about those two types).

In terms of precision, recall and F1-score, XceptionNet model achieves the highest F1-score corresponding to "rainbow" category with 97%, while the worst F1-score is related to "snow" category

with only 69% which is the same result of VGG16 model. However, the best F1-score of VGG16 model corresponds to category "lightening" with 98%. The "lightening" category is the one with the best F1-score for ResNet50V2, InceptionV2 and DensNet201 with 98%, 99% and 100%, respectively. The worst F1-score of ResNet50V2 model corresponds to "frost" and "sandstorm" with 75%, while the worst category in case of InceptionV3 is "frost" with 81%. Finally, DenseNet201 worst case corresponds to "rime" category with 82% as F1-score.

4.5. Discussion of the Fusion Models Results

Fig. 3 proves that categories "fogsmog" and "sandstorm" which have similar textures cause 8 FN errors for score-level fusion model, but only three errors in case of meta-based fusion models, which leads to a conclusion that the meta-based fusion enhanced the classification accuracy by minimizing errors caused by texture similarity. The same conclusion is true in case of "rime" and "glaze" which cause 14 FN errors in case of score-level fusion, 6 errors in case of meta-based RF fusion, and 6 errors in case of meta-based Bagging fusion. The accuracy of many categories is improved using fusion method, especially the meta-fused methods. For example, in term of meta-based fusion model, the categories "lightening", and "rainbow" have no FP nor FN errors which are not registered by any of the individual models. For fusion models Score-level fusion: The highest F1-score is 100, which corresponds to the categories "hail," "Lightning," "Rainbow," and "rain." For Meta-based fusion, the highest F1-score is 100%, which corresponds to the categories "hail," "Lightning," "Rainbow," and "rain." Meta-based Bagging fusion, the highest F1-score is 100%, which corresponds to the categories "hail," "Lightning," "Rainbow," and "rain". Table 3 includes a comparison of the current study and previous ones in the field of weather image classification.

Table 3. Comparison with the current state-of-art methodologies

Researcher	Methodology	Dataset	Results	Main Limitations
Wang et al. [16]	Multi-task learning methodology with DenseNet and ResNet	Multi-class weather dataset of 9 climates	68.25% ResNet50, 72.25% ResNet101, 72.75% DenseNet	Not all conditions considered; low accuracy
Galeb et al. [17]	CNN with simple Machine Learning models (DT, SVM)	Kaggle dataset of 1500 images in 5 categories	92% CNN, 93%, CNN+DT, 94% CNN+SVM	Small dataset size and number of categories
Kalkan et al. [18]	Transfer learning models (VGG16, MobileNet, ResNet152, DenseNet	Dataset of cloudy and not cloudy images	91.4% accuracy with VGG16	Only two classes are considered (cloudy or clear); no fusion
Çetiner [19]	Transfer learning; used ResNet152V2 model	WEAPD dataset	88% accuracy	Low accuracy
Alhaija et al. [10]	Various deep learning models (SqueezeNet, ResNet-50, EfficientNet)	A subset of 1656 weather images, 6 categories	96.05%SqueezeNet, 98.48% ResNet, 97.78%EfficientNet	No fusion or it was used to enhance the performance
Mashudi et al. [20]	InceptionResNetV2, XceptionNet, MobileNet,DenseNet201	WEAPD dataset	83% as the best performance using DensNet201	No fusion or ensemble used
Current study	Meta-based RF fusion of many deep models	WEAPD dataset (11 categories)	Accuracy 96% Precision, recall, and F1-score: 96%	The dataset size is somehow low

Table 3 illustrates that the current study outperforms all previous studies that worked on the same WEAPD dataset. The current study got benefit of the meta-based RF fusion methodology to improve the performance. Besides that, the current study took into account all possible weather conditions which were not addressed by previous studies.

4.6. Limitations

The current study solved as much as possible of previous studies' problems. The results obtained by this study outperformed all previous ones applied to the same dataset and this was due to the implemented enhancements. However, there are still some limitations that future studies can override.

Since the dataset includes 11 categories, some categories contain a small number of images. Although the current methodology applied some data augmentation tasks, the next studies can use other techniques like oversampling. There are still experiments that can be applied using different DL models like attention-based and transformer-based vision models. Future work can benefit of our study and try to generalize the current methodologies by utilizing different datasets and compare to the findings. Other implementations can focus on the problem of similarities between some weather conditions or other special weather conditions that were not considered in this study. Hyperparameters tuning is not addressed in the current study and can be performed by future tries.

5. Conclusion

In this study, a new meta-based RF fusion of many transfer deep learning models for weather classification problem was introduced. The used models are XceptionNet, VGG16, ResNet50V2, InceptionV3 and DenseNet201. The models were trained and tested on the WEAPD dataset consisting of 11 weather classes. Score-level fusion and meta-based Bagging fusion were also used and compared with the meta-based RF fusion model. In the pre-processing step, the data augmentation methods were applied to increase images and improve the training process. For training, the "Adam" optimizer and "categorical cross-entropy" loss function were used. An early stop condition of patience factor equal to 6 was used, and the dataset was split into 80% for training and 20% for testing. A validation set of 20% of the training set is also utilized. The results show that the ResNet50V2 model achieved the best performance among the individual models, with a weighted F1-score of 90%. This indicates that ResNet50V2 was able to accurately classify the various weather types based on images. The other four models achieved lower weighted F1-scores of 80%. However, the meta-based and score-level fusion methods combining the predictions of the different models improved the classification performance, achieving a weighted F1-score of up to 96%. This indicates that meta-based fusion can help improve the accuracy of weather classification from images compared to individual models alone. Besides, the meta-based fusion outperformed the score-level fusion. Future studies can focus on other data oversampling techniques to increase the training size and treat the class-imbalance problem of categories like SMOTE and other oversampling methods. Besides that, researchers can investigate datasets with a larger number of images.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] D. M. J. S. Bowman, "Detecting, Monitoring and Foreseeing Wildland Fire Requires Similar Multiscale Viewpoints as Meteorology and Climatology," *Fire*, vol. 6, no. 4, p. 160, Apr. 2023, doi: 10.3390/fire6040160.

[2] M. H. Younus and R. Mohammed, "Geo-informatics techniques for detecting changes in land use and land cover in response to regional weather variation," *Theor. Appl. Climatol.*, vol. 154, no. 1–2, pp. 89–106, Oct. 2023, doi: 10.1007/s00704-023-04536-8.

[3] C.-A. D. Tsiakos and C. Chalkias, "Use of Machine Learning and Remote Sensing Techniques for Shoreline Monitoring: A Review of Recent Literature," *Appl. Sci.*, vol. 13, no. 5, p. 3268, Mar. 2023, doi: 10.3390/app13053268.

[4] M. Toğaçar, B. Ergen, and Z. Cömert, "Detection of weather images by using spiking neural networks of deep learning models," *Neural Comput. Appl.*, vol. 33, no. 11, pp. 6147–6159, Jun. 2021, doi: 10.1007/s00521-020-05388-3.

[5] M. Guzel, M. Kalkan, E. Bostanci, K. Acici, and T. Asuroglu, "Cloud type classification using deep learning with cloud images," *PeerJ Comput. Sci.*, vol. 9, p. e1779, Jan. 2024, doi: 10.7717/peerj-cs.1779.

[6] Y. Liu, X. Huang, and D. Liu, "Weather-Domain Transfer-Based Attention YOLO for Multi-Domain Insulator Defect Detection and Classification in UAV Images," *Entropy*, vol. 26, no. 2, p. 136, Feb. 2024, doi: 10.3390/e26020136.

[7] S. Dalal, B. Seth, M. Radulescu, T. F. Cilan, and L. Serbanescu, "Optimized Deep Learning with Learning without Forgetting (LwF) for Weather Classification for Sustainable Transportation and Traffic Safety," *Sustainability*, vol. 15, no. 7, p. 6070, Mar. 2023, doi: 10.3390/su15076070.

[8] K. U. Jaseena and B. C. Kovoor, "Deterministic weather forecasting models based on intelligent predictors: A survey," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3393–3412, Jun. 2022, doi: 10.1016/j.jksuci.2020.09.009.

[9] M. Chen, J. Sun, K. Aida, and A. Takefusa, "Weather-aware object detection method for maritime surveillance systems," *Futur. Gener. Comput. Syst.*, vol. 151, pp. 111–123, Feb. 2024, doi: 10.1016/j.future.2023.09.030.

[10] F. Q. Kareem, A. M. Abdulazeez, and D. A. Hasan, "Predicting Weather Forecasting State Based on Data Mining Classification Algorithms," *Asian J. Res. Comput. Sci.*, vol. AJRCOS, no. 3, pp. 13–24, Jun. 2021, doi: 10.9734/ajrcos/2021/v9i330222.

[11] N. Rai *et al.*, "Applications of deep learning in precision weed management: A review," *Comput. Electron. Agric.*, vol. 206, p. 107698, Mar. 2023, doi: 10.1016/j.compag.2023.107698.

[12] M. M. Taye, "Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions," *Computers*, vol. 12, no. 5, p. 91, Apr. 2023, doi: 10.3390/computers12050091.

[13] Q. A. Al-Haija, M. Gharaibeh, and A. Odeh, "Detection in Adverse Weather Conditions for Autonomous Vehicles via Deep Learning," *AI*, vol. 3, no. 2, pp. 303–317, Apr. 2022, doi: 10.3390/ai3020019.

[14] H. Xiao, F. Zhang, Z. Shen, K. Wu, and J. Zhang, "Classification of Weather Phenomenon From Images by Using Deep Convolutional Neural Network," *Earth Sp. Sci.*, vol. 8, no. 5, p. e2020EA001604, May 2021, doi: 10.1029/2020EA001604.

[15] B. Bochenek and Z. Ustrnul, "Machine Learning in Weather Prediction and Climate Analyses—Applications and Perspectives," *Atmosphere (Basel).*, vol. 13, no. 2, p. 180, Feb. 2022, doi: 10.3390/ATMOS13020180/S1.

[16] Y. Wang and Y. Li, "Research on Multi-class Weather Classification Algorithm Based on Multi-model Fusion," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Jun. 2020, pp. 2251–2255, doi: 10.1109/ITNEC48623.2020.9084786.

[17] M. Ghaleb, H. Moushier, H. Shedeed, and M. Tolba, "Weather Classification using Fusion Of Convolutional Neural Networks and Traditional Classification Methods," *Int. J. Intell. Comput. Inf. Sci.*, vol. 22, no. 2, pp. 1–13, May 2022, doi: 10.21608/ijicis.2022.117060.1156.

[18] M. Kalkan *et al.*, "Cloudy/clear weather classification using deep learning techniques with cloud images," *Comput. Electr. Eng.*, vol. 102, p. 108271, Sep. 2022, doi: 10.1016/j.compeleceng.2022.108271.

[19] H. Çetiner and S. Metlek, "Classification of Weather Phenomenon with a New Deep Learning Method Based on Transfer Learning," *Int. Conf. Recent Acad. Stud.*, vol. 1, no. 1, pp. 92–99, May 2023, doi: 10.59287/icras.678.

[20] N. A. Mashudi, N. Ahmad, S. M. Sam, N. Mohamed, and R. Ahmad, "Deep Learning Approaches for Weather Image Recognition in Agriculture," in *2022 IEEE Symposium on Future Telecommunication Technologies (SOFTT)*, Nov. 2022, pp. 72–77, doi: 10.1109/SOFTT56880.2022.10010161.

[21] H. Xiao, "Weather phenomenon database (WEAPD)," *Harvard Dataverse*, 2021. [Online]. Available at: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/M8JQCR.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. (ICLR 2015)*, Apr. 2015. [Online]. Available at: http://www.robots.ox.ac.uk/.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9908 LNCS, Springer Verlag, 2016, pp. 630–645, doi: 10.1007/978-3-319-46493-0_38

[24] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, vol. 2017-Janua, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, vol. 2016-Decem, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[26] J. Lai *et al.*, "[A DenseNet-based diagnosis algorithm for automated diagnosis using clinical ECG data]," *Nan Fang Yi Ke Da Xue Xue Bao*, vol. 39, no. 1, pp. 69–75, 2019. [Online]. Available at: https://pubmed.ncbi.nlm.nih.gov/30692069/.

[27] M. Steurer, R. J. Hill, and N. Pfeifer, "Metrics for evaluating the performance of machine learning based automated valuation models," *J. Prop. Res.*, vol. 38, no. 2, pp. 99–129, Apr. 2021, doi: 10.1080/09599916.2020.1858937.

[28] M. Z. Naser and A. H. Alavi, "Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences," *Archit. Struct. Constr.*, vol. 3, no. 4, pp. 499–517, Dec. 2023, doi: 10.1007/s44150-021-00015-8.

[29] A. M. Elkhatat, K. Elsaid, and S. Almeer, "Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text," *Int. J. Educ. Integr.*, vol. 19, no. 1, p. 17, Sep. 2023, doi: 10.1007/s40979-023-00140-5.

[30] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: Multi-Label Confusion Matrix," *IEEE Access*, vol. 10, pp. 19083–19095, 2022, doi: 10.1109/ACCESS.2022.3151048.

[31] N. S. Ranawat, J. Prakash, A. Miglani, and P. K. Kankar, "Performance evaluation of LSTM and Bi-LSTM using non-convolutional features for blockage detection in centrifugal pump," *Eng. Appl. Artif. Intell.*, vol. 122, p. 106092, Jun. 2023, doi: 10.1016/j.engappai.2023.106092.

[32] D. Božić, B. Runje, D. Lisjak, and D. Kolar, "Metrics Related to Confusion Matrix as Tools for Conformity Assessment Decisions," *Appl. Sci.*, vol. 13, no. 14, p. 8187, Jul. 2023, doi: 10.3390/app13148187.

[33] S. Khozama and A. M. Mayya, "A New Range-based Breast Cancer Prediction Model Using the Bayes' Theorem and Ensemble Learning," *Inf. Technol. Control*, vol. 51, no. 4, pp. 757–770, Dec. 2022, doi: 10.5755/j01.itc.51.4.31347.

[34] J. T. Hancock, T. M. Khoshgoftaar, and J. M. Johnson, "Evaluating classifier performance with highly imbalanced Big Data," *J. Big Data*, vol. 10, no. 1, p. 42, Apr. 2023, doi: 10.1186/s40537-023-00724-5.