Iran Journal of Computer Science (2025) 8:491–509
https://doi.org/10.1007/s42044-025-00230-2

RESEARCH

Enhanced ore classification through optimized CNN ensembles and feature fusion

Mustafa Yurdakul[1] · **Kübra Uyar**[2] · **Şakir Taşdemir**[3]

Received: 11 November 2024 / Accepted: 3 January 2025 / Published online: 29 January 2025

Abstract

Ore is a type of natural stone that contains economically valuable minerals or metals. Accurate classification of ore minerals is crucial for improving operational efficiency in mining, reducing environmental impacts, and determining market value. Traditional methods for classifying ores are often time-consuming, labor-intensive, and error-prone. Therefore, computer-aided systems offer a significant advantage in this field. In this study, various efficient Deep Learning (DL) approaches are utilized for the detection of ore types. Within the scope of the study, four different experiments (transfer learning, feature extraction and classification with SVM, feature selection with optimization algorithms, and ensemble methods) are conducted, and the methods are compared in terms of classification metrics. As a result of the experimental case studies, high accuracy rates between 95 and 98% are achieved. The most successful method is the ensemble method, weighted by grid search. The ensemble model, which combined AlexNet, VGG16, and Xception models, achieves remarkable results with an overall accuracy of 98.11%, precision of 98.18%, recall of 98.11%, and f1-score of 98.11% on the publicly available Ore Images Dataset (OID). This study demonstrates that efficient DL approaches can classify ores with very high accuracy and have significant potential applications in the mining industry.

Keywords CNN · Ensemble learning · Feature fusion · Feature selection · Optimization · Ore classification

1 Introduction

Ores are naturally occurring mineral resources that contain economically valuable metals or elements. These resources play a significant role in many industries such as metallurgy, construction, and energy. Ores are critical not only for their economic yield but also for sustainable industrial practices. However, increasing demand on a global scale is leading to the depletion of high-quality ore deposits and increasing the need to process lower-quality resources. This not only raises economic concerns but also environmental impacts. The correct classification of ores is an important step in optimizing mining and processing processes. Accurate classification reduces the environmental burden by reducing energy and resource consumption during mineral processing, while at the same time enabling more efficient utilization of valuable minerals. However, this process presents significant challenges with current approaches. Conventional methods are often based on manual observation or limited analytical methods and fail to accurately distinguish complex mineral compositions. Nowadays, with the development of image-based classification and automation technologies, DL methods offer new opportunities in this field. In particular, Convolutional Neural Networks (CNNs) have emerged as a promising solution for ore classification due to their superior ability to extract features from complex visual data.

CNN is an effective and robust DL algorithm for image analysis studies such as classification, detection, and segmentation problems in various fields [1–3]. Several techniques have been proposed to improve the classification performance of CNN models, such as using transfer learning,

✉ Kübra Uyar
kubra.uyar@alanya.edu.tr

Mustafa Yurdakul
mustafayurdakul@kku.edu.tr

Şakir Taşdemir
stasdemir@selcuk.edu.tr

1 Computer Engineering Department, Kirikkale University, Kırıkkale, Turkey

2 Computer Engineering Department, Alanya Alaaddin Keykubat University, Antalya 07425, Turkey

3 Computer Engineering Department, Selcuk University, Konya, Turkey

CNN-based feature extraction, feature fusion, feature selection, and ensemble. Kim et al. [4] used VGG16 for defect detection in manufacturing, obtaining total positive rates (TPR) of 42.40% to 99.43% when trained from scratch. TPRs improved from 99.80% to 100% with transfer learning. Li et al. [5] proposed a multimodel cascaded convolutional neural network (MC-CNN) combining three subnetworks (DSSD, YOLOv4, and Faster-RCNN) for domestic waste image detection and classification. Experimental results indicated a performance, with an average improvement of 10% in detection precision. Yang et al. [6] evaluated GoogleNet and AlexNet models for the classification of glioma tumors in magnetic resonance images using both scratch training and transfer learning. While the test accuracy of AlexNet was 85.50% in scratch training, this rate increased to 92.70% when transfer learning was used. Similarly, GoogleNet's 90.90% accuracy with scratch training increased to 94.5% with transfer learning. Öztel et al. [7] compared AlexNet and VGG16 for facial expression recognition. AlexNet achieved 84% accuracy from scratch and 95% with transfer learning. VGG16 reached 16.67% from scratch and 98.33% with transfer learning. The CNNs were also used for feature extraction, and the extracted features were classified using various ML algorithms. Özaltın and Yeniay [8] proposed a novel CNN model to categorize electrocardiogram signals and they achieved a f1-score of 96.53% and an accuracy of 96.53%. The model was also used as a feature extractor and classified the features with a support vector machine (SVM), achieving an f1-score of 99.20% and an accuracy of 99.21%. Oğuz et al. [9] proposed a CNN model for glaucoma detection from optical coherence tomography images. The model achieved an f1-score of 88.35% and an accuracy rate of 86.62%. They used SVM, k-nearest neighbors (kNN), random forest, AdaBoost, multilayer perceptrons, and Naive Bayes (NB) for classification. The AdaBoost algorithm obtained the best result with a f1-score of 93.75% and an accuracy of 92.96%. In the literature, feature fusion has been used to combine features extracted from more than one CNN model. Keerthana et al. [10] used AlexNet, GoogleNet, VGG16, VGG19, ResNet18, ResNet50, ResNet101, ShuffleNet, MobileNet, and DenseNet201 for skin cancer image classification. DenseNet201 achieved the highest f1-score of 92.83% and an accuracy of 85.63%, while MobileNet ranked second. They fused the features extracted by MobileNet and DenseNet201 and classified the features with SVM. They achieved a f1-score of 93.10% and an accuracy of 88.02%. The use of CNNs as feature extractors increases performance; however, some features may cause a longer training time, generate noise, and decrease classification performance. Selecting the most representative features from the features extracted by CNN is an optimization problem. For this purpose, various optimization algorithms have been used in the literature. Aslan et al. [11] extracted texts as word representations using the FastText word embedding approach to classify tweets posted during the COVID-19 pandemic as positive, negative, and neutral. Then, each word representation was converted into a 19 x 300 image. A novel CNN model was proposed to classify the tweet images, which achieved a f1-score of 86.52% and an accuracy of 89.71%. Furthermore, the model was used as a feature extractor and the extracted features were selected by arithmetic optimization algorithm. Then the features were classified by SVM, NB, Logistic Regression (LR), decision trees, and kNN algorithms. The most successful result was achieved with a f1-score of 93.71% and an accuracy of 95.09% by the kNN model. Kundu and Chattopadhyay [12] employed GoogLeNet and ResNet18 models to classify cervical precancerous cells. In the Mendeley LBC dataset, GoogleNet and ResNet18 achieved accuracies of 82.90% and 91.19%, respectively. Similarly, in the SIPaKMeD 2-class dataset, these models achieved accuracies of 92.96% and 90.49%, respectively. Finally, on the SIPaKMeD 5-class dataset, GoogleNet and ResNet18 achieved accuracies of 85.50% and 85.13%, respectively, demonstrating their effectiveness in classifying cervical precancerous cells in various datasets. Additionally, the models were used as feature extractors and the extracted features were combined. To select optimal features, different optimization algorithms were employed and features were classified with an SVM classifier.

Some literature studies that use DL methods exist for the classification task of ore types. Liu et al. [13] compared current mainstream instance segmentation models such as FCIS, SOLO, Mask R-CNN, and YOLACT + + and determined that YOLACT + + serves as the base network. On the test set, the accuracy of the coke and sintered ore granularity recognition models was achieved at 96.78%. Sun et al. [14] designed an efficient DL framework named LosNet for ore segmentation in real-time. To increase the speed and diminish the computational complexity of the model, lightweight feature pyramid networks were used. Zhou et al. [15] proposed a novel CNN model that combines data augmentation, transfer learning, and SENet for automated ore classification. MobileNet, which uses fewer parameters than existing network models, was offered for a fast calculation speed. They achieved an accuracy of 96.89% with a designed hybrid approach that automatically extracts features. Chen et al. [16] designed a low-resolution borehole image stitching framework that automates the analysis of geotechnical borehole images and allows field safety analysts to efficiently handle irregularities in real time. They proposed an improved vision transformer with an algebraic multigrid model for image feature recognition, collecting borehole datasets from different construction sites containing ore, coal, oil, and gas. Liu et al. [17] worked on small DL models evaluating and optimizing

Table 1 Comparative summary of ore classification studies in the literature

Reference study	Year	Methodology
[14]	2022	LosNet, a lightweight feature pyramid network for efficient DL framework
[15]	2022	CNN with data augmentation, transfer learning, and SENet; MobileNet for fast calculation speed
[17]	2021	Analysis of the effects of model depth, structure, and dataset size on ore classification accuracy
[18]	2020	Deployment of VGG16 model and data augmentation techniques
[19]	2019	Application of CNN beyond binary classification to address ore sorting problems
[20]	2011	3D images to identify different kinds of ores and direct process inputs

model depth, model structure, and data size for the ore classification task. The effect of model depth, model structure, and, dataset size on the classification accuracy of mineral images was analyzed using various datasets. Olivier et al. [18] studied the estimation of ore particle size distribution using deep CNN. VGG16 was deployed and data augmentation approaches were used to improve the performance of the model. After calculating the prediction error for each size class per image (10 prediction errors of each of the 118 test images produces 1180 prediction error values), they obtained a mean error is -0.012 and a standard deviation is 0.107. Karimpouli and Tahmasebi [19] studied to improve the segmentation process of digital rock images. They extended the potential of CNN beyond binary classification by solving ore sorting problems. Singh et al. [20] developed a method that can detect the area, face, or part of the face that might not be a particular type of ore and have the surface coating over a particular side using 3D images. It was suggested that all types of iron ore can be identified as the basis of one or the other property. Therefore, it is easy to identify different kinds of ores by image processing techniques, and any intelligent system can direct the process inputs using this technique. Table 1 lists the comparative analysis of our study with literature studies mentioned in the literature review that use a similar dataset.

According to Table 1, it is seen that there are many studies on the classification of ore types in the literature. In existing studies, classification is based on a single model, which often struggles to fully capture the diversity and complexity of the dataset. It is not always possible for a single model to capture the different features of all classes equally well. Particularly in cases where the amount of data is insufficient and imbalanced, the model may overfit the majority classes and not learn the rare or more difficult classes well enough. Consequently, the performance of the model in the face of different conditions, new examples, or real-life variations is degraded, limiting its ability to generalize. In the proposed work, effective DL methods are used to solve these problems. In Case #1, the most successful CNN models are determined, in Case #2, the features extracted by these models are fused and classified, in Case #3, feature selection is performed to find some discriminating features using meta-heuristic algorithms, and in Case #4, the ensemble method is tested. In these approaches, multiple models contribute to the classification process, providing more robust solutions to problems faced by individual models, such as overfitting, poorly learned classes due to small data, and low generalization ability.

Considering the gaps in the literature, the main outputs and contributions of this study can be summarized as follows.

- In this study, OID consisting of 957 images of 7 different ore types is used. This dataset is an up-to-date and public source for ore type classification.
- The main problem that this study aims to solve is the overfitting, low generalization ability, and insufficient learning of some classes in the classification of ore types. To solve these problems, the hypothesis "Combining multiple models, feature fusion, meta-heuristic based feature selection and ensemble learning improves generalization ability and reduces overfitting" is proposed.
- Reviewed and analyzed in detail recent studies on ore type classification and CNN performance improvements, identifying the main trends and gaps in the existing literature.
- Various advanced techniques such as transfer learning, feature fusion, feature selection, and ensemble learning are tested and their impact on ore type classification is evaluated.
- Seventeen advanced CNN models with different topologies (e.g., dense, inception, residual, separable convolution) are tested for ore classification and the best-performing models are determined.
- After identifying the most successful models, we extract and fuse the features from these models to create a more comprehensive feature set and classify them with SVM algorithm. The SVM algorithm is tested with different kernels to determine the best kernel.
- After the fusion, various optimization algorithms are used to select the most relevant features from the feature set. The selected features are then classified with SVM.
- Finally, combined the three best-performing CNN models using a weighted voting ensemble, which allowed us to improve the overall classification accuracy and robustness by utilizing the strengths of each model.

The results obtained in our study show that our proposed methods have high accuracy and generalization capacity. The ensemble learning approach yields the best classification result with 98.11% accuracy, 98.18% precision, 98.11% recall, 98.11% f1-score, 99.92% AUC, and 97.80% Cohen's Kappa, and outperformed other literature studies.

The rest of this paper is organized as follows. Section 2 introduces the materials and methods used in the experiments. Section 3 provides an in-depth description of the experimental case studies methodologically. Section 4 presents the experimental results and corresponding discussions. The conclusions of the study are summarized in Sect. 5.

2 Material and methods

The proposed methods are based on an investigation of deep techniques to improve the classification performance of ore types. The basic knowledge about the concept of CNN with transfer learning, the combination of extracted CNN features named feature fusion, optimization approaches to select discriminative features of CNN models after feature fusion operation, ensemble of CNNs for robust classification models, performance evaluation metrics, and the description of the ore dataset used in the experimental cases are explained in detail in the following subsections.

2.1 CNNs

CNN is a DL algorithm inspired by the human visual cortex [21] and is generally used for image analysis tasks. CNN has shown remarkable results in various fields such as medicine, defense, autonomous driving systems, agriculture, and face recognition. In the experimental studies, seventeen state-of-the-art CNN models that contain various topologies are tested. These include residual blocks (ResNet50, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2), MobileNet and MobileNetV2 models which are recognized for their lightweight and efficient design, and NASNetMobile with a customized structure, stacked blocks (AlexNet, VGG16, VGG19), dense blocks (DenseNet121, DenseNet169, DenseNet201), and inception module (InceptionV3), and separable convolution model (Xception). These CNNs were trained individually with the ore images for multiclass classification.

2.2 Transfer learning and fine-tuning

Transfer learning is the process of applying knowledge and experience gained in one domain to a different domain. Typically, a CNN model is first trained on a large dataset to learn general features and patterns in the input image. Then the model is retrained on a different dataset, which contains less data. During the second training phase, the model acquires specific features of the dataset in addition to the basic features learned from the first dataset. This results in more rapid and robust classification with less data. All experimental CNN models used in this study are pre-trained on the ImageNet dataset. The final layers of the tested CNN models, which include a classification layer designed to classify ImageNet data into 1000 classes, are modified and tuned depending on the class number of the ore dataset which is 7. To classify seven different types of ore images, the final layers of the models are replaced with a Global Average Pooling (GAP) layer and a softmax layer with seven outputs.

2.3 Ensemble learning

Ensemble learning is a method that utilizes the predictions of multiple models to make more accurate predictions. It is used to obtain a model with high generalization ability by combining the strengths of different models. There are various ensemble learning methods, including bagging, boosting, stacking, and voting. Voting ensemble is among one of the most commonly used techniques in literature [22]. Voting ensemble approaches can be implemented using various aggregation strategies, such as hard, soft, or weighted voting, allowing flexibility in how predictions are combined to optimize predictive performance based on the specific characteristics of the problem domain. Soft voting involves taking the probabilistic predictions of different classifiers and averaging them to determine the final decision. The mathematical formula for the soft voting process is given in Eq. 1.

$$SoftVoting = \frac{1}{N}\sum_{i=1}^{N} P_i(x) \tag{1}$$

where $p_i(x)$ represents the prediction of the i-th classifier and N corresponds to the total number of classifiers. On the other hand, hard voting involves taking the class predictions from each classifier and accepting the class with the most votes as the final decision. The mathematical formula for the hard voting method is given in Eq. 2.

$$HardVoting = mod(y_1, y_2, y_3, \ldots, y_N) \tag{2}$$

where y_i represents the class prediction of the i-th classifier. Finally, weighted voting is used where each model's prediction is given a weight and the weighted predictions are summed to determine the final class with the highest value. The mathematical formula for the weighted voting process is given in Eq. 3.

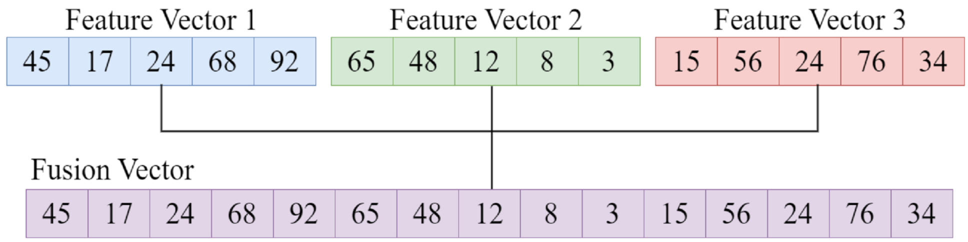

Fig. 1 Schematic diagram of feature fusion: Feature vector 1, 2, and 3 represent different feature sets and the fusion vector represents the concatenation of all feature vectors into a single feature vector

$$WeightedVoting = \sum_{i=1}^{N} w_i * y_i \tag{3}$$

where w_i represents the weight of the i-th model and y_i implies the prediction of that model. The class with the highest total value is selected as the final decision. The determination of weights for the prediction of each model is a challenging optimization problem. In the related experimental case, three CNN models with the best results according to the classification results are selected and weights are determined by the grid search algorithm so that the sum of the weights is one.

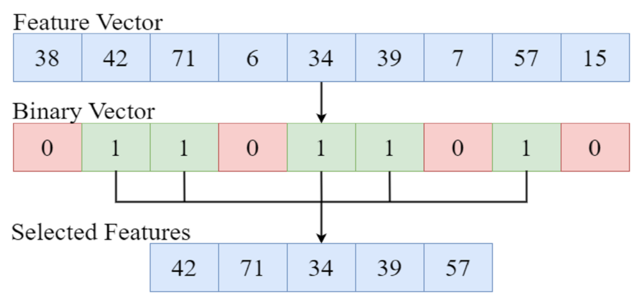

Fig. 2 Schematic diagram of feature selection with binary vector: The feature vector represents the raw features, while the binary vector is a mask used for feature selection. Selected features are the intersection of the feature vector and the binary vector; it consists only of features with a value of 1 in the binary vector

2.4 Feature extraction with CNN and fusion of features

In ML techniques, it is necessary to extract and manually choose features for classification, however, this approach requires expert knowledge of domain-specific information. In particular, for complex data patterns, it becomes increasingly difficult to extract the features manually. On the other hand, with DL methods, the features are automatically learned by the model. As a result, deep feature extraction and classification yield higher accuracy results than typical ML techniques. It is possible to use the domain-specific knowledge that different CNN models have collected to create a class-discriminative generic feature set by using the feature fusion approach [23, 24].

For the experimental studies, the best-performing three CNN models among seventeen models are used as feature extractors. A schematic diagram of the fusion is shown in Fig. 1. The GAP layer is added during the transfer learning phase to be used as a feature extractor layer. The features are then fused and classified using SVM models using linear, RBF, polynomial, and sigmoid kernels.

2.5 Feature selection using optimization approaches

The fusion of deep features makes classification performance more robust, however, some features may cause noise and affect performance negatively. The feature selection of features plays a crucial role in identifying the optimal subset of variables that effectively characterize the original dataset. Feature selection using optimization algorithms is a powerful approach to improve the performance of models by identifying the most relevant features from a dataset. Optimization algorithms provide a more sophisticated mechanism to navigate the vast search space of possible feature subsets. These algorithms aim to maximize a fitness function while minimizing the number of selected features, striking a balance between model simplicity and predictive power. The flexibility of optimization algorithms enables them to handle non-linear relationships, interactions between features, and constraints specific to the problem domain.

The selection of meaningful features can be implemented with a binary optimization approach whether the given feature is selected or not. Given N features, a vector of binary values of size N can be used to determine which features to use. A binary value of 0 indicates that the feature will be excluded, whereas a binary value of 1 corresponds to the feature that will be included. Optimization starts with a population of candidate solutions or a single solution. The algorithm iteratively updates these solutions to find the subset that maximizes the fitness function. The feature selection process with binary vector is shown in Fig. 2.

In this study, different optimization models based on evolutionary which are the Genetic Algorithm (GA) and Memetic Algorithm (MA), nature inspired which are Artificial Bee Colony (ABC) and Flower Pollination Algorithm

Table 2 The category and working principle details of the optimization algorithms used in the study

Optimization approach	Category	Working principle
ABC [25]	Swarm	Simulates the foraging behavior of honeybees, where each bee searches for promising solutions representing potential feature subsets
GA [26]	Evolutionary	Inspires the principles of natural selection and genetics, and applies crossover and mutation to evolve to the best feature set
FPA [27]	Nature-inspired	Inspires the natural pollination process of flowers and has a robust global search capacity to efficiently explore a wide range of feature combinations
MA [28]	Evolutionary	Integrates the global exploration power of GA with local search strategies for fine-tuning, striking a balance between improving known solutions and discovering new solutions
PSO [29]	Swarm	Converges toward the best solution through information sharing between particles, mimicking the social behavior of birds or fish

(FPA), and swarm intelligence which is the Particle Swarm Optimization (PSO) are used for feature selection. Table 2 lists the details of the optimization algorithms. We have selected these algorithms based on their time of emergence and popularity. Each algorithm offers a unique strategy to overcome the feature selection optimization challenge, aiming to reduce noise and increase the robustness of classification performance.

Fig. 3 Sample images from the dataset: representative images of seven different ore types

2.6 Dataset description

The OID [15], which is publicly available on the Kaggle platform and contains a total of 957 images, was used in the experimental studies. The dataset contains images of seven different ore types: biotite, bornite, chrysocolla, malachite, muscovite, pyrite, and quartz. The distribution of dataset classes is given in Table 3 and sample images from the OID are illustrated in Fig. 3.

The classification performance of CNN is directly proportional to the amount of data available, however, it is not always possible to obtain sufficient data. In such cases, data augmentation techniques can be used to increase the amount of data available. In this work, the ore images were augmented using some methods such as center cropping, edge cropping, zooming, and brightness transformation as used in [15]. Experimental studies were started by increasing the data so that the number of images in all classes was distributed equally.

2.7 Performance measurement metrics

In experimental studies, a variety of measures are used to evaluate the classification performances of deep approaches. Accuracy is used to measure the overall accuracy of the model, representing the ratio of True Positive (TP) and True Negative (TN) results to the total number of cases. Precision is used to calculate the percentage of cases identified as positive (true positive—TP) that are positive. Recall is used to measure the accuracy with which TPs are identified. F1-score is a balanced combination of precision and recall, taking into account both false positives (FP) and false negatives (FN). Additionally, the ability of the model to discriminate between classes is assessed using the Area Under the Curve (AUC) and Cohen's kappa score. The mathematical formulas for these performance measurement metrics are given in Eqs. 4–8. The

Table 3 The distribution of classes in the dataset and a total number of images

Biotite	Bornite	Chrysocolla	Malachite	Muscovite	Pyrite	Quartz	Total
68	170	164	235	77	98	145	957

values of TP, TN, FP, and FN are calculated using the confusion matrices.

$$Accuracy = (TP + TN)/(TP + FN + FP + TN) \quad (4)$$

$$Precision = TP/(TP + FP) \quad (5)$$

$$Recall = TP/(TP + FN) \quad (6)$$

$$F1 - score = (2x Precision x Recall)/ (Precision + Recall) \quad (7)$$

$$Cohen's Kappa = (P_0 - P_e)/(1 - P_e) \quad (8)$$

In Eq. 8, P_0 is defined as the observed proportion of agreement, which represents the relative frequency of cases where the raters agree. P_e is the expected proportion of agreement, calculated on the basis of the frequency of each category, and represents the probability of random agreement between the raters.

3 Background of the experimental case studies

The proposed study is based on the investigation of techniques to improve the ore type classification performances. Deep and ensemble methods improve the performance of the classification models. In addition, regularization methods such as transfer learning, fine-tuning, and data augmentation techniques affect the overall performance of the deep models positively.

In the proposed study, four different situations are evaluated. In the first case study, Case #1, firstly, the performances of well-known pre-trained CNN models in ore classification are analyzed. Then, as in Case #2, taking into account the effective feature extraction features of CNN models, the features extracted by the best-performing three CNN models as a result of Case #1 are combined and the feature fusion step is completed. On the other hand, considering that not every extracted feature contains meaningful information and that distinctive features have an impact on performance, meaningful and distinctive features are selected for the image using optimization approaches in Case #3. Finally, the three CNN models with the best classification performance (Case #1) are ensembled for utilizing the predictions of multiple models to make more accurate predictions. Graphical representation of the experimental cases is illustrated in Fig. 4. As a result of these experimental case studies, the effects of various deep models and the effect of approaches such as fusion and ensemble implemented on these models on performance were analyzed.

4 Experimental results and discussion

4.1 Results

In this section, the results of the experiments are detailed and discussed. Finally, a comparison between the proposed method with existing literature is given and the reliability of the proposed approach is established. The overall performance of all of the CNN models and SVM models used in this study was evaluated using a fivefold cross-validation. The data set was divided into five equal parts, with each part used sequentially as a test set, and the model trained on the remaining four parts. The performance of the model was evaluated for each fold, and this process was repeated five times. Finally, the overall performance of the model was assessed by averaging the five values obtained, which provides a more comprehensive and reliable assessment.

For a fair comparison, considering the previous study [15] that uses the same ore dataset, CNN models were trained for 50 epochs using the Adam optimizer with a learning rate of 0.0001 and a cross-entropy loss function. The optimization algorithms were executed for 100 iterations. The choice of 100 iterations is based on several factors. First, meta-heuristic algorithms do not guarantee finding the global optimum; however, their goal is to find a near-optimal solution. The number of iterations can vary depending on the complexity of the problem and the algorithm used. 100 iterations provide a reasonable balance between computational effort and solution quality and allow the algorithm to explore the search space extensively.

The experiments were conducted on a computer equipped with 128 GB RAM, an Intel(R) Core i9-10920X CPU @3.50 GHz, and an NVIDIA GeForce RTX 3060 graphics card. The implementation was performed using Python (v3.9.18), Keras (v2.10.0), and Scikit-Learn (v1.3.2) libraries.

In this section, four different experiment case studies were performed to demonstrate the best approach to classify ore types. In Case #1, seventeen state-of-the-art CNN models

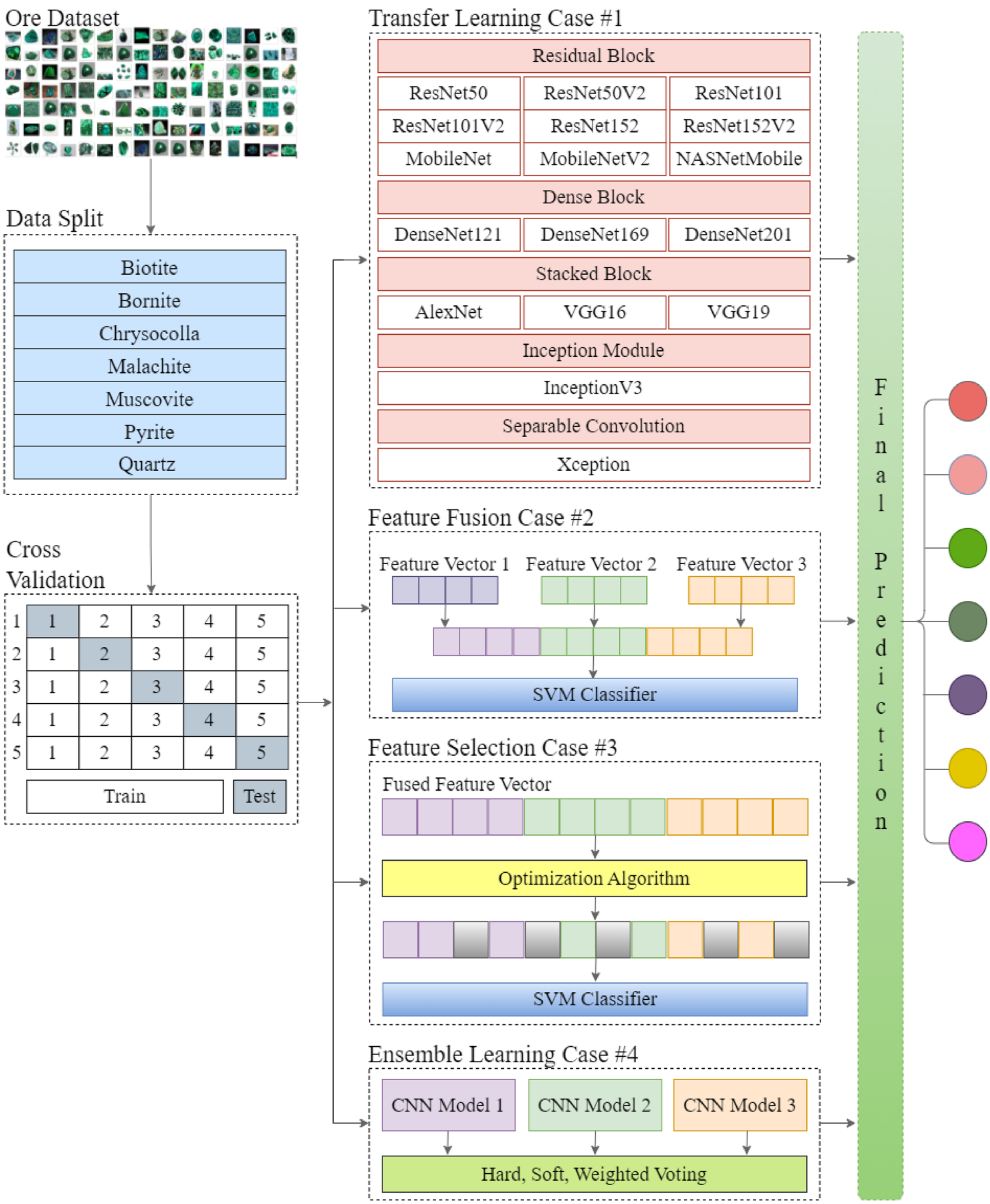

Fig. 4 Graphical abstract of the experimental cases to improve the classification performances of ore types

were tested with transfer learning and fine-tuning. In Case #2, extracted features from the best-performing three CNN models were fused and classified with SVM. In Case #3, five different meta-heuristic algorithms were used to select the most distinctive and meaningful ones of the fused features. In Case #4, the best-performing three CNN models were ensembled with various voting methods.

4.1.1 Case #1: Transfer learning

In this experiment, AlexNet, DenseNet, InceptionV3, MobileNet, NASNet, ResNet, VGG, and Xception models were used as a backbone to classify ore types. All models were initialized on ImageNet weights using transfer learning and retrained on the OID. Accuracy, precision, recall, f1-score, Cohen's kappa, and AUC metrics were used to objectively and accurately evaluate the classification performance of the models on the test data.

The performance results are presented in Table 4, showing that the tested CNN models achieved classification accuracies ranging from 76 to 95%. Among them, AlexNet stood out with the highest accuracy at 95.93%, accompanied by 96.10% precision, 95.93% recall, a 95.93% f1-score, a 99.48% AUC value, and a Cohen's kappa of 95.25%. In the DenseNet series, both DenseNet121 and DenseNet169 achieved 91.79% accuracy, but DenseNet201 performed best within that series, reaching 94.10% accuracy, 94.66% precision, 94.10% recall, a 94.19% f1-score, a 99.68% AUC, and a Cohen's kappa of 93.12%. InceptionV3 achieved 93.98% accuracy, MobileNet and MobileNetV2 had accuracies of 90.88% and 85.35%, respectively, and NASNetMobile showed the lowest performance with 76.29% accuracy, 83.55% precision, 76.29% recall, a 75.35% f1-score, a 96.85% AUC, and a 72.34% Cohen's kappa. The ResNet series models produced very similar results, with ResNet50 leading that group at 91.98% accuracy. VGG16 and VGG19 were similarly effective, achieving 95.68% and 94.10% accuracy, while Xception performed well at 94.22% accuracy, 94.54% precision, 94.22% recall, a 94.20% f1-score, a 99.48% AUC, and a 93.26% Cohen's kappa. All models demonstrated consistent performance, indicated by low standard deviation values.

Overall, AlexNet emerged as the top-performing model, with VGG16 and Xception following closely. In contrast, NASNetMobile's results were the weakest. Notably, AlexNet's simpler architecture contributed to its strong performance, particularly given the limited amount of training data available. While larger, more complex models often need substantial amounts of data to properly adjust their weights, AlexNet was able to make the most of a smaller dataset, achieving effective learning without the need for large-scale training sets. This characteristic gives AlexNet a clear advantage over more complex architectures when working with limited data.

Figure 5 shows the mean accuracy and mean loss values of the three most successful models during training. In addition to this, colored shadow parts demonstrate the standard deviation of the accuracy and loss values during training. According to Fig. 5, it is seen that there is no significant difference between the accuracy and loss values in both training and validation data. Therefore, models are not overfitting and have a high generalization capability.

The complexity matrices in Fig. 6 show the class-based classification performances of AlexNet, DenseNet, InceptionV3, MobileNet, NASNetMobile, ResNet, VGG, and Xception models in detail. It is especially noteworthy that AlexNet, InceptionV3, VGG16, VGG19, and Xception models make excellent predictions in some classes. AlexNet, InceptionV3, and Xception models accurately classified the biotite class with 47 correct predictions. The DenseNet and ResNet series models performed consistently overall, with a low number of misclassifications in the bornite and chrysocolla classes. MobileNet and MobileNetV2 struggled to correctly predict the chrysocolla and malachite classes. In contrast, the VGG16 and VGG19 models performed consistently with high prediction rates in most classes overall but made minor errors in the muscovite and pyrite classes. The Xception model was one of the most consistent models and performed particularly well in the biotite, bornite, muscovite, pyrite, and quartz classes. As a result, AlexNet, VGG16, VGG19, and Xception models achieved the most successful results in class-based analysis. On the other hand, NASNetMobile and MobileNetV2 models made more errors in some classes, indicating that their performance is limited. This situation is directly related to the number of parameters, architectural complexity, and learning capacity of the models [17, 30].

4.1.2 Case #2: Feature fusion results

This section details the classification results obtained by fusing the features extracted from the three most successful models. AlexNet, VGG16, and Xception models, which obtained the best classification results after transfer learning, were used as feature extractors. The reason for using three models during feature extraction is that the increase in classification performance would not be sufficient if two models were used and model diversity would be limited. If more than three models were used, the computational load would increase. Therefore, three models were used to increase performance while keeping the computational load at a manageable level.

AlexNet, VGG16, and Xception models extracted 256, 512, and 2048 features, respectively. These features are combined side by side and given and used as input to the SVM algorithm for the classification process. The SVM classifier was tested using linear, poly, sigmoid, and RBF kernel functions. These kernel functions were chosen because of their suitability for different data distributions and class separation properties. In this way, it was aimed to find the most appropriate kernel function. Table 5 presents the classification performance results obtained using each of these kernel functions in detail.

Table 5 shows the impact of the kernel functions used in the SVM classifier on the classification performance. The Poly kernel exhibited a low accuracy rate of 80.97%, with precision of 88.39%, recall of 80.97%, f1-score of 82.18%, AUC of 98.78%, and Cohen's kappa of 77.80%. The main reason for this is that the Poly kernel creates too much complexity in high-dimensional feature sets and can lead to overfitting problems. Sigmoid and RBF kernels achieved similar performance with 91% and 91.12% accuracy, precision of 91.25%

Table 4 Comparative performance results of CNN models

Model	Performance metrics (%)							
	Accuracy	Precision	Recall	F1-score	AUC	Cohen's kappa	Lowest accuracy	Highest accuracy
AlexNet	**95.93 (± 0.0593)**	**96.10 (± 0.0564)**	**95.93 (± 0.0593)**	**95.93 (± 0.0592)**	**99.48 (± 0.0102)**	**95.25 (± 0.0692)**	**84.19**	**1.000**
DenseNet121	91.79 (± 0.0387)	92.79 (± 0.0265)	91.79 (± 0.0387)	91.69 (± 0.0410)	99.48 (± 0.0036)	90.43 (± 0.0451)	84.50	9.544
DenseNet169	91.79 (± 0.0378)	92.47 (± 0.0295)	91.79 (± 0.0378)	91.78 (± 0.0378)	99.45 (± 0.0018)	90.43 (± 0.0441)	85.11	96.35
DenseNet201	94.10 (± 0.0155)	94.66 (± 0.0102)	94.10 (± 0.0155)	94.19 (± 0.0139)	99.68 (± 0.0011)	93.12 (± 0.0180)	91.19	95.74
InceptionV3	93.98 (± 0.0134)	94.16 (± 0.0119)	93.98 (± 0.0134)	93.96 (± 0.0134)	99.49 (± 0.0022)	92.98 (± 0.0156)	92.10	95.44
MobileNet	90.88 (± 0.0281)	91.38 (± 0.0239)	90.88 (± 0.0281)	90.66 (± 0.0308)	99.47 (± 0.0026)	89.36 (± 0.0327)	86.63	94.53
MobileNetV2	85.35 (± 0.0339)	87.91 (± 0.0222)	85.35 (± 0.0339)	85.18 (± 0.0351)	98.61 (± 0.0056)	82.91 (± 0.0396)	82.67	91.79
NASNetMobile	76.29 (± 0.0693)	83.55 (± 0.0263)	76.29 (± 0.0693)	75.35 (± 0.0764)	96.85 (± 0.0169)	72.34 (± 0.0809)	66.26	86.63
ResNet101	90.58 (± 0.0246)	91.96 (± 0.0158)	90.58 (± 0.0246)	90.49 (± 0.0260)	99.00 (± 0.0048)	89.01 (± 0.0287)	87.23	93.92
ResNet101V2	90.09 (± 0.0280)	90.81 (± 0.0246)	90.09 (± 0.0280)	90.03 (± 0.0286)	98.89 (± 0.0046)	88.44 (± 0.0327)	85.11	93.01
ResNet152	91.67 (± 0.0495)	92.70 (± 0.0355)	91.67 (± 0.0495)	91.52 (± 0.0524)	99.18 (± 0.0071)	90.28 (± 0.0578)	82.37	95.74
ResNet152V2	91.61 (± 0.0325)	92.68 (± 0.0196)	91.61 (± 0.0325)	91.80 (± 0.0296)	99.28 (± 0.0039)	90.21 (± 0.0380)	85.71	95.14
ResNet50	91.98 (± 0.0176)	92.47 (± 0.0145)	91.98 (± 0.0176)	91.93 (± 0.0186)	99.09 (± 0.0011)	90.64 (± 0.0205)	89.36	94.22
ResNet50V2	91.85 (± 0.0317)	92.30 (± 0.0281)	91.85 (± 0.0317)	91.84 (± 0.0317)	99.36 (± 0.0028)	90.50 (± 0.0369)	87.84	96.35
VGG16	95.68 (± 0.0089)	95.78 (± 0.0090)	95.68 (± 0.0089)	95.68 (± 0.0089)	99.70 (± 0.0004)	94.96 (± 0.0104)	94.22	96.96
VGG19	94.10 (± 0.0045)	94.29 (± 0.0039)	94.10 (± 0.0045)	94.11 (± 0.0045)	99.53 (± 0.0016)	93.12 (± 0.0053)	93.62	94.83
Xception	94.22 (± 0.0155)	94.54 (± 0.0132)	94.22 (± 0.0155)	94.20 (± 0.0154)	99.48 (± 0.0027)	93.26 (± 0.0181)	91.79	96.66

and 91.35%, recall of 91% and 91.12%, f1-score of 90.96% and 91.11%, AUC of 99.29% and 99.27%, and Cohen's kappa of 89.50% and 89.64%, respectively. These two functions could not fully demonstrate their performance due to noise and redundant features in the fused feature set.

The Linear kernel was the best classifier of the dataset, achieving the highest accuracy (93.13%). However, even this result was below the performance of the individual CNN models in Case #1. The size of the fused feature set and redundant features negatively affected the generalization ability of the SVM classifier. The redundant features caused noise, resulting in lower accuracy rates. To reduce these negative effects, the feature selection techniques in Case #3 were applied.

Figure 7 shows the confusion matrices of the SVM classifier using the combined features with different kernel functions. A common challenge across all models is that the malachite class is often misclassified as the chrysocolla class. It indicates that the feature representations of these two classes overlap significantly, leading to uncertainties in classification. However, as can be seen from the complexity matrices, the linear, RBF and sigmoid kernels are more successful than the Poly kernel in terms of overall classification consistency.

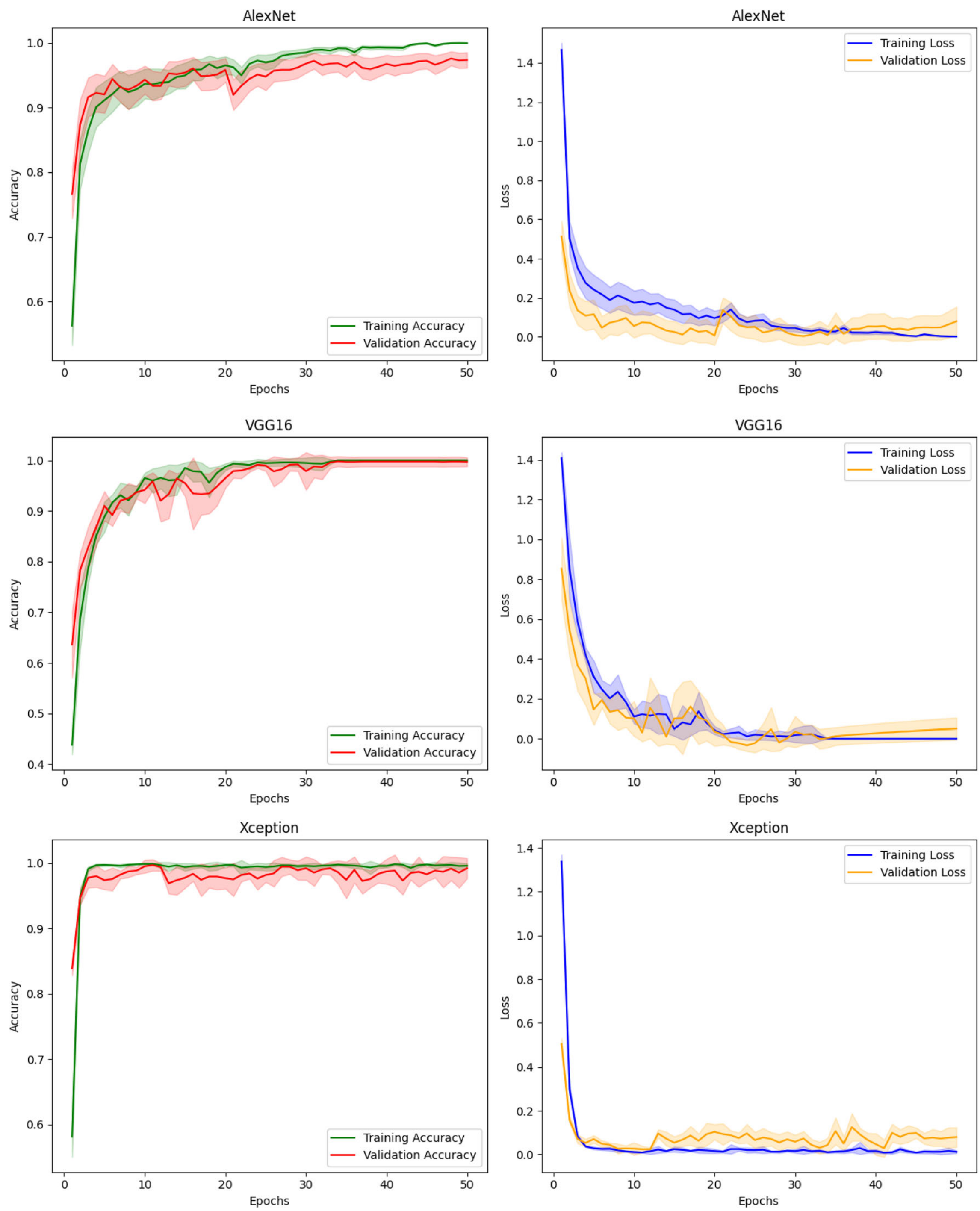

Fig. 5 Average minibatch accuracy and loss values of the best-performing three CNN models during training. The red and green lines represent the average value of the metric across the five folds, while the shadows of the color around it indicate the standard deviation value

4.1.3 Case #3: Feature selection results

In Case #1, AlexNet, VGG16, and Xception outperformed with 95.93%, 95.68%, and 94.22% accuracy, respectively. In Case #2, fused features were classified with SVM and achieved an accuracy rate, but this value is lower than the performance of the individual models. It indicates that the large and complex set of fused features has a negative impact on the generalization capability. In particular, the presence of redundant or repetitive features created a burden on the classifier, leading to a performance loss. In this context, the need for feature selection has clearly emerged. In this experiment,

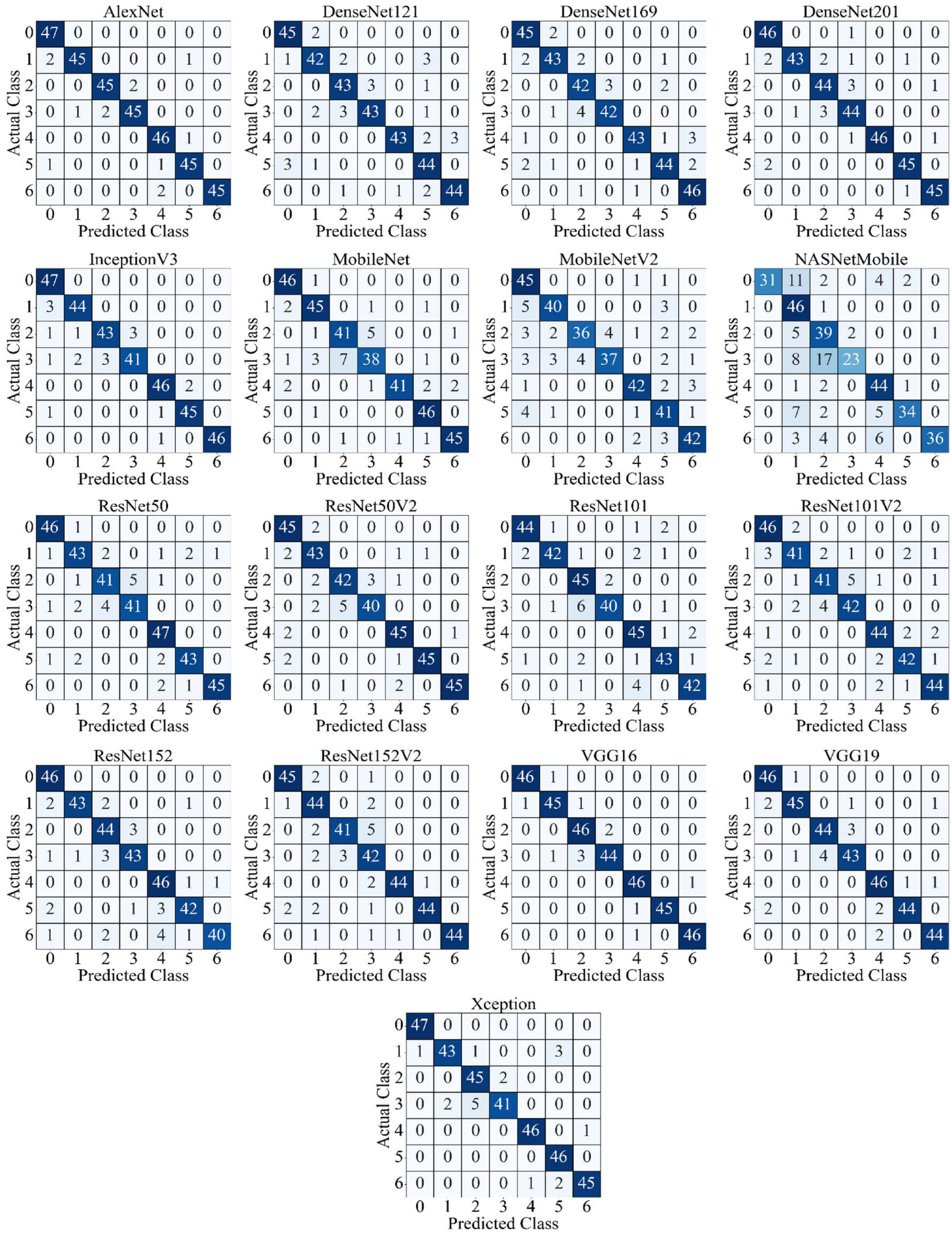

Fig. 6 Confusion matrices of CNN models (0:Biotite, 1:Bornite:, 2:Chrysocolla, 3:Malachite, 4:Muscovite, 5:Pyrite, 6:Quartz)

Table 5 Comparative performance results of the SVM classifier on the fused feature vector using different kernel types

SVM kernel	Performance metrics (%)							
	Accuracy	Precision	Recall	F1-score	AUC	Cohen's kappa	Lowest accuracy	Highest accuracy
Linear	**93.13 (± 0.0152)**	**93.31 (± 0.0146)**	**93.13 (± 0.0152)**	**93.08 (± 0.0156)**	**99.45 (± 0.0016)**	**91.98 (± 0.0178)**	90.88	94.83
Poly	80.97 (± 0.0454)	88.39 (± 0.0045)	80.97 (± 0.0454)	82.18 (± 0.0350)	98.78 (± 0.0044)	77.80 (± 0.0529)	74.16	86.01
Sigmoid	91.00 (± 0.0185)	91.25 (± 0.0185)	91.00 (± 0.0185)	90.96 (± 0.0191)	99.29 (± 0.0024)	89.50 (± 0.0216)	88.44	93.67
RBF	91.12 (± 0.0193)	91.35 (± 0.0187)	91.12 (± 0.0193)	91.11 (± 0.0196)	99.27 (± 0.0030)	89.64 (± 0.0226)	88.44	93.31

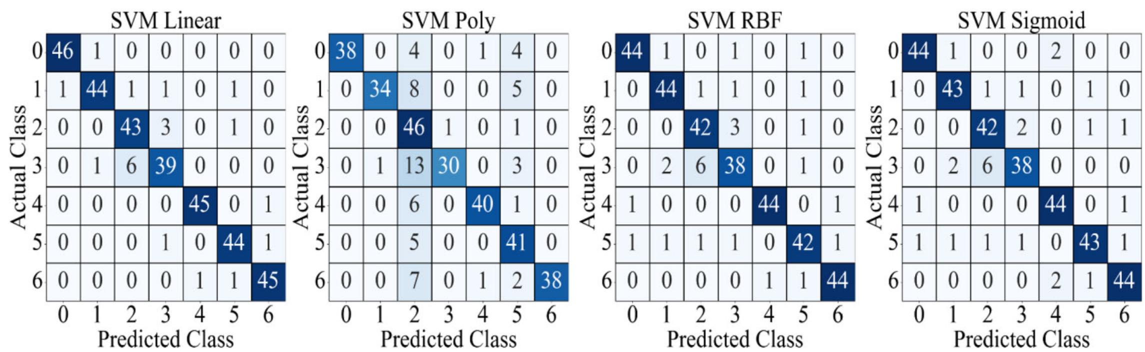

Fig. 7 Confusion matrices of SVM models with different kernels (0:Biotite, 1:Bornite, 2:Chrysocolla, 3:Malachite, 4:Muscovite, 5:Pyrite, 6:Quartz)

it is aimed to remove unnecessary features that negatively affect the classification performance from the fused feature set. For this purpose, a smaller and more meaningful feature subset was obtained to improve the classifier performance. In this study, from the comprehensive feature set obtained from the three best performing CNN models and combined to 2816 features, the most discriminative subset of features was selected by meta-heuristics and classified using SVM with Linear kernel, the most successful kernel determined as a result of Case#2. In this process, ABC, GA, FPA, MA, and PSO algorithms were trained for 100 iterations and reached optimal results with solutions containing 622, 566, 477, 558, and 1455 features, respectively. The f1-score value was used as the fitness function of the optimization algorithms.

Table 6 presents the classification performance results of the optimization algorithms that incorporated feature selection. These results indicate that using meta-heuristic algorithms for feature selection significantly improved classification outcomes. Among them, the ABC and FPA algorithms both achieved the highest accuracy rate of 96.29%. While ABC recorded a precision of 96.43%, recall of 96.29%, f1-score of 96.27%, AUC of 99.56%, and Cohen's kappa of 95.67%, FPA's corresponding values were 96.42% precision, 96.29% recall, 96.28% F1-score, 99.58% AUC, and 95.67% Cohen's kappa. Despite their close results, FPA emerged as the most successful feature selection algorithm when considering all metrics together, outperforming ABC. In contrast, the PSO algorithm lagged behind the top performers, achieving a 95.25% accuracy, 95.40% precision, 95.25% recall, 95.24% f1-score, 99.56% AUC, and a 94.46% Cohen's kappa. These results show that redundant features in the large merged feature set negatively affect the classification performance and that feature selection methods improve accuracy rates by obtaining smaller, meaningful subsets. Overall, feature selection in this process reduced the burden on the classifier and improved its generalization capability. The results in Table 6 show that the meta-heuristic algorithms exhibit low standard deviation values for all performance metrics, indicating that the methods produce consistent and reliable results.

The complexity matrices in Fig. 8 show the class-based performance of the meta-heuristic algorithms. It can be observed that the overall classification performance is high for all algorithms. Biotite, bornite, chrysocolla, muscovite, pyrite, and quartz classes have very high correct classification rates. However, the malachite class is particularly noteworthy as a class with relatively higher misclassification rates. Malachite is often confused with the chrysocolla class, indicating that the characteristics of these two classes overlap

Table 6 Comparative performance results of the optimization algorithms used for feature selection

Algorithm	Performance metrics(%)							
	Accuracy	Precision	Recall	F1-score	AUC	Cohen's kappa	Lowest accuracy	Highest accuracy
ABC	96.29 (± 0.0119)	**96.43 (± 0.0114)**	96.29 (± 0.0119)	96.27 (± 0.0118)	99.56 (± 0.0019)	95.67 (± 0.0139)	94.22	97.56
GA	95.98 (± 0.0111)	96.14 (± 0.0105)	95.98 (± 0.0111)	95.98 (± 0.0110)	99.56 (± 0.0017)	95.31 (± 0.0129)	94.52	97.26
FPA	**96.29 (± 0.0116)**	96.42 (± 0.0115)	**96.29 (± 0.0116)**	**96.28 (± 0.0116)**	**99.58 (± 0.0017)**	**95.67 (± 0.0135)**	**94.52**	**98.17**
MA	96.10 (± 0.0088)	96.27 (± 0.0089)	96.10 (± 0.0088)	96.10 (± 0.00887)	99.58 (± 0.0019)	95.46 (± 0.0103)	94.52	96.96
PSO	95.25 (± 0.0073)	95.40 (± 0.0071)	95.25 (± 0.0073)	95.24 (± 0.0072)	99.56 (± 0.0018)	94.46 (± 0.0085)	94.22	96.35

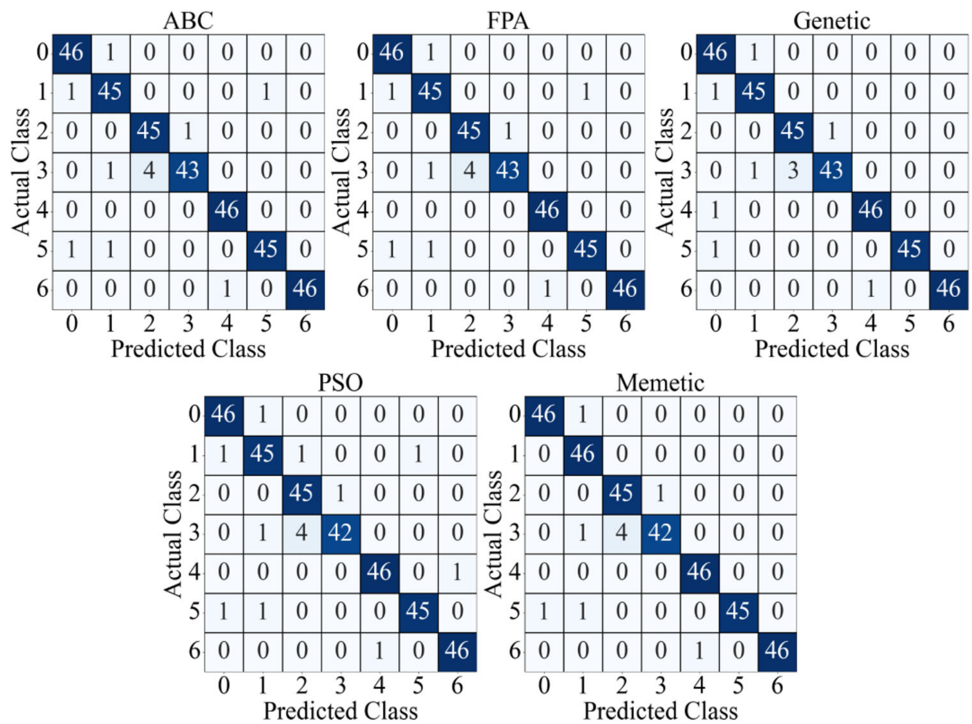

Fig. 8 Confusion matrices of meta-heuristic-based selected features and SVM (0:Biotite:, 1:Bornite:, 2:Chrysocolla, 3:Malachite, 4:Muscovite, 5:Pyrite, 6:Quartz)

significantly. Especially in the PSO algorithm, 4 misclassifications were observed in the malachite class. In contrast, the ABC and FPA algorithms have a lower misclassification rate. In general, ABC and FPA algorithms have better class-based consistency than the other algorithms, which is supported by their low error rates.

4.1.4 Case #4: Ensemble learning results

As seen in Case #3, feature selection using optimization approaches on the fused features improved the classification performance of ore images. Moreover, to evaluate the performance of the ensemble approach that improves the predictive performance of models by combining multiple models together, this case study was carried out. For this purpose, the three CNN models (AlexNet, VGG16, and Xception) that performed the best in Case #1 were selected and their predictions were combined to perform classification. The reason for choosing three models is to find an optimal balance between computational complexity and classification performance.

As an ensemble method, hard voting, soft voting, and weighted voting strategies were used. The hard voting method is based on the majority of the class predictions given by each model. The soft voting method combines the probability predictions of each model and selects the class with the

Table 7 Comparative performance results of various voting ensemble techniques

Model	Performance metrics(%)							
	Accuracy	Precision	Recall	F1-score	AUC	Cohen's kappa	Lowest accuracy	Highest accuracy
Hard	97.39 (± 0.0119)	97.51 (± 0.0116)	97.39 (± 0.0119)	97.38 (± 0.0119)	**99.94 (± 0.0010)**	96.95 (± 0.0139)	95.14	98.48
Soft	97.69 (± 0.0105)	97.82 (± 0.0099)	97.69 (± 0.0105)	97.68 (± 0.0104)	99.94 (± 0.0010)	97.30 (± 0.0122)	95.74	98.78
Weighted	**98.11 (± 0.0231)**	**98.18 (± 0.0222)**	**98.11 (± 0.0231)**	**98.11 (± 0.0231)**	99.92 (± 0.0013)	**97.80 (± 0.0270)**	**93.61**	**99.69**

Table 8 Ensemble weights of the CNN models (w1: AlexNet, w2: VGG16, w3: Xception)

Weight 1 (w1)	Weight 2 (w2)	Weight 3 (w3)
0.5	0.1	0.4

highest probability. In weighted voting, the predictions were combined using weights determined according to the performance of each model. The models were used with hard, soft, and weighted voting ensemble methods, and the performance results are presented in Table 7.

Table 7 indicates that the weighted voting method performed best across nearly all evaluation criteria, except for AUC. With weighted voting, the model achieved 98.11% accuracy, 98.18% precision, 98.11% recall, a 98.11% f1-score, a 99.92% AUC, and a Cohen's kappa of 97.80%. In comparison, the hard voting approach reached 97.39% accuracy, 97.51% precision, 97.39% recall, 97.38% f1-score, 99.94% AUC, and 96.95% Cohen's kappa. The soft voting approach followed closely with 97.69% accuracy, 97.82% precision, 97.69% recall, 97.68% f1-score, 99.94% AUC, and 97.30% Cohen's kappa. To determine the optimal weights for each model in the weighted voting method, a grid search algorithm was employed, ensuring the weights were summed to one.

Table 8 shows the weights assigned to the AlexNet, VGG16, and Xception models. These weights represent the contribution of each model in the ensemble method and its importance in the decision process. With the highest weight value of 0.5, the AlexNet model has a more dominant role in the ensemble method compared to the other two models. It is related to the fact that AlexNet's individual classification performance (as shown in Table 4) is quite good. On the other hand, the VGG16 model contributed the least to the decision process with a weight value of 0.1. The Xception model, with a weight value of 0.4, was positioned somewhere between AlexNet and VGG16. The distribution of these weights was determined to optimize the generalization capability and classification performance of the ensemble method by exploiting the strengths of each model.

Some differences are observed in the class-based performances of hard voting, soft voting, and weighted voting methods as shown in Fig. 9. The hard and soft voting methods performed similarly in general, but misclassified 3 samples as chrysocolla in the Malachite class. In contrast, the weighted voting method performed better in this class with only 1 misclassification. All methods achieved high accuracy rates in the biotite, bornite, chrysocolla, muscovite, pyrite, and quartz classes, but the malachite class stood out with more errors than the other classes.

In this study, four different experimental approaches were used to classify ore types using DL techniques, and their highest classification performances are summarized in Table 9. In Case #1, AlexNet achieved 95.93% accuracy, 96.10% precision, 95.93% recall, a 95.93% f1-score, a 99.48% AUC, and a 95.25% Cohen's kappa. VGG16 followed closely with 95.68% accuracy, 95.78% precision, 95.68% recall, a 95.68% f1-score, a 99.70% AUC, and a 94.96% Cohen's kappa. Xception attained 94.22% accuracy, 94.54% precision, 94.22% recall, a 94.20% f1-score, a 99.48% AUC, and a 93.26% Cohen's kappa. In Case #2, the features extracted from the three successful models were fused and then classified with an SVM using a linear kernel. This approach achieved a 93.13% accuracy, 93.31% precision, 93.13% recall, a 93.08% f1-score, a 99.45% AUC, and a 91.98% Cohen's kappa. While the results were still strong, the performance was not as high as the best models from the transfer learning phase, indicating that fusing features introduced some redundancy or noise. In Case #3, using the FPA algorithm to select meaningful features raised the accuracy to 96.29%, with 96.42% precision, 96.29% recall, a 96.28% f1-score, a 99.58% AUC, and a 95.67% Cohen's kappa. These results demonstrated that carefully removing redundant features improves classification performance. In Case #4, combining AlexNet, VGG16, and Xception models through weighted voting led to 98.11% accuracy, 98.18% precision, 98.11% recall, a 98.11% f1-score, a 99.92% AUC,

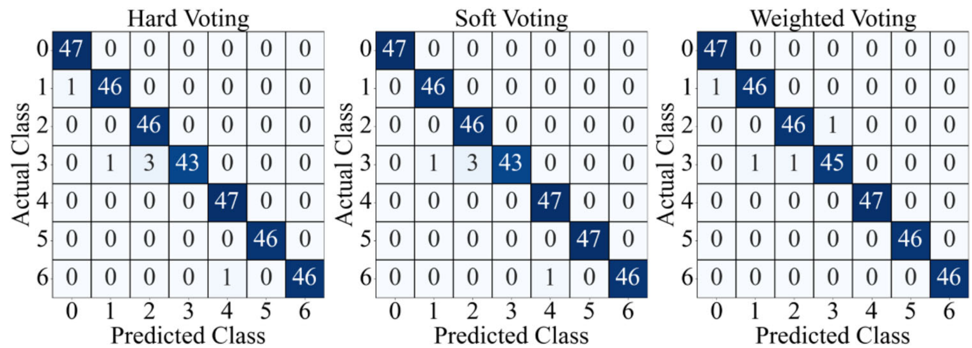

Fig. 9 Confusion matrices of ensemble techniques (0:Biotite, 1:Bornite:, 2:Chrysocolla, 3:Malachite, 4:Muscovite, 5:Pyrite, 6:Quartz)

Table 9 Comparative performance results of the four cases tested in the study

Case No	Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC (%)	Cohen's kappa (%)
Case #1	Transfer learning and fine-tuning	95.93 (± 0.0593)	96.10 (± 0.0564)	95.93 (± 0.0593)	95.93 (± 0.0592)	99.48 (± 0.0102)	95.25 (± 0.0692)
Case #2	Feature fusion	93.13 (± 0.0152)	93.31 (± 0.0146)	93.13 (± 0.0152)	93.08 (± 0.0156)	99.45 (± 0.0016)	91.98 (± 0.0178)
Case #3	Feature selection with optimization algorithm	96.29 (± 0.0116)	96.42 (± 0.0115)	96.29 (± 0.0116)	96.28 (± 0.0116)	99.58 (± 0.0017)	95.67 (± 0.0135)
Case #4	**Ensemble learning**	**98.11 (± 0.0231)**	**98.18 (± 0.0222)**	**98.11 (± 0.0231**	**98.11 (± 0.0231)**	**99.92 (± 0.0013)**	**97.80 (± 0.0270)**

and a 97.80% Cohen's kappa. This ensemble approach surpassed all previous methods by leveraging the strengths of each individual model, creating a more powerful and generalizable classification tool.

4.1.5 Case #5: Comparison of weighted-voting ensemble with other literature studies

In the literature, there are many studies on the classification of ore types using DL techniques. However, most of these studies were conducted on different datasets, which makes it difficult to make a direct comparison. The only study on the OID was conducted by [15].

Within the scope of this study, four different methods (cases) were tested and it was determined that the most successful results were obtained with the weighted voting ensemble method. In Table 10, the classification performances of the proposed weighted voting ensemble method and the other study are presented comparatively. The proposed method improves the accuracy rate from 96.89% to 98.11%, which is about a 2% performance improvement over the literature using the same dataset. Considering the market value of ores and the critical importance of classification accuracy in industrial applications, this improvement is quite valuable.

Zhou et al. [15] used a MobileNet model improved with the attention mechanism, however, this model only performs classification with a single backbone. On the other hand, the proposed ensemble method is a combination of AlexNet, VGG16, and Xception models. The advantage of this approach is that different models can perform strongly on different classes. For example, some models struggle to distinguish certain classes, while other models can successfully classify the same classes. This variety increases the generalization capability of the ensemble method and provides a more consistent performance. As a result, the proposed ensemble method has made a significant contribution to the literature by achieving better classification results than other work in the literature.

4.2 Discussion

Ore classification is important for assessing quality and determining the most appropriate processing methods in mining and material sciences. In this study, four different efficient DL approaches were used to classify ore types. In the first experiment, seventeen CNN models were used with transfer learning. The three most successful models were AlexNet, VGG16, and Xception with accuracy rates of 95.93%, 95.68%, and 94.22%, respectively (Table 4). For

Table 10 Comparison of the proposed method with a literature study that uses the same dataset

Reference study	Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC (%)	Cohen's kappa (%)
Zhou et al. [15]	MobileNet + Transfer Learning + Data Augmentation + Squeeze-and-Excitation Networks	96.89	–	–	–	–	–
Proposed study (Case #4)	Weighted-Voting Ensemble of CNNs (AlexNet, VGG16, Xception)	**98.11**	**98.18**	**98.11**	**98.11**	**98.11**	**97.80**

this classification problem, it is obvious that when the model goes deeper, overfitting issues occur and the generalization ability for ore classification tasks reduces. In the second experiment, the best-performing three CNN models were used as feature extractors and the extracted features were fused. Fused features were classified using linear, poly, sigmoid, and RBF kernels of the SVM. Linear kernel achieved a 93.13% accuracy rate (Table 5). In the third experiment, some meaningful and discriminative features in combined features were selected using ABC, GA, FPA, MA, and PSO algorithms and classified with SVM. As a result of the selection of features using optimization approaches, FPA achieved a 96.29% accuracy rate (Table 6). FPA surpassed other optimization algorithms not only in its classification accuracy but also in using fewer features for the final feature set. In the last experiment, the best-performing three CNN models were ensembled with hard, soft, and weighted voting methods. In Case #4, a 98.11% accuracy was achieved with the weighted voting method with weights determined by grid search. As a result, the most successful result was obtained with the weighted voting approach with an accuracy rate of 98.11% (Table 7). The ensemble approach that reduces variance and improves to generalization of the proposed method outperformed the work in the literature using the same dataset (Table 10).

5 Conclusion

Ores are valuable minerals in mining and materials sciences, and accurately classifying them is essential for increasing operational efficiency and mitigating environmental impacts. While traditional classification methods tend to be time-consuming and prone to human error, DL approaches can automate this process, providing faster and more accurate results. This study aimed to determine the most effective DL method for classifying ore types by examining four different approaches. Initially, several pre-trained CNN models were evaluated using the OID. AlexNet reached an accuracy of 95.93%, while VGG16 and Xception followed closely with 95.68% and 94.22%, respectively. The precision values for these models were similarly high—96.10% for AlexNet, 95.78% for VGG16, and 94.54% for Xception—as were their recall and f1-scores, all hovering around or above 94%. Furthermore, the AUC values exceeded 99%, and Cohen's kappa values approached or surpassed 93%, indicating consistently excellent performance. In the second stage, the features extracted from the final layers of the three successful models were combined and then classified using an SVM with a linear kernel. After feature fusion, the classification accuracy reached 93.13%. While these values remain strong, there has been a noticeable decline compared to individual models. This decline was attributed to noise by redundant features during the fusion process. To address this issue, the third stage employed various optimization algorithms (ABC, GA, FPA, MA, and PSO) to eliminate redundant features and retain only the most meaningful ones. Among these, the FPA algorithm achieved the best results, increasing the accuracy to 96.29%. Not only did FPA outperform the other algorithms in terms of accuracy, but it also accomplished this with fewer features, thereby streamlining the classification process. Finally, the fourth experiment combined AlexNet, VGG16, and Xception using an ensemble learning approach. Three voting methods (hard, soft, and weighted) were tested. Weighted voting yielded the best results, achieving an accuracy of 98.11%. These results surpassed the performance of all previous experiments as well as those reported in the existing literature. The proposed ensemble model not only reduced variance but also enhanced generalization capability. Overall, this method provides more robust, consistent, and generalizable outcomes than relying on any single model alone.

As a conclusion, the proposed ensemble approach is beneficial for mining and material experts in the decision of early detection of ore minerals. Moreover, as tested in experimental cases in the results section, the deep ensemble approach can be widely adopted to solve various problems due to its stability, reliability, and robustness. However, there are some limitations of this work, such as the fact that the OID contains only seven ore types. The diversity of the dataset is limited, which may lead to limitations in real-life applications. However, with the cooperation of professionals in the mining industry, the data diversity can be increased. In addition, training optimization algorithms is time-consuming. These processes can be accelerated by using cloud-based solutions, parallel computing methods or powerful hardware.

Author contributions Mustafa Yurdakul, Kübra Uyar: Conceptualization, Methodology, Review and editing, Software, Validation, Visualization, Writing-original draft, Şakir Taşdemir: Conceptualization, Methodology, Supervision.

Funding The authors did not receive support from any organization for the submitted work.

Data availability The data on which the experiments are carried out are publicly available on Kaggle Platform (https://www.kaggle.com/asiedubrempong/minerals-identification-dataset) and appropriate citations are added that will certainly help in reproducibility of this work.

Code availability The source code used to obtain experimental results is publicly available at https://github.com/ymyurdakul/Ore-Classification-with-Deep-CNN-Models/tree/main.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article. The authors declare no competing interests.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Consent of publication Not applicable.

References

1. Farhadipour, A.: Lung and colon cancer detection with convolutional neural networks and adaptive histogram equalization. Iran J. Comput. Sci. **7**(2), 381–395 (2024). https://doi.org/10.1007/s42044-023-00161-w
2. Wijaya, N., Mulyani, S.H., Anggraini, Y.W.: DeepFruits: efficient citrus type classification using the CNN. Iran J. Comput. Sci. **6**(1), 21–27 (2023). https://doi.org/10.1007/s42044-022-00117-6
3. Doğan, G., Ergen, B.: A new approach based on convolutional neural network and feature selection for recognizing vehicle types. Iran J. Comput. Sci. **6**(2), 95–105 (2023). https://doi.org/10.1007/s42044-022-00125-6
4. Kim, S., Kim, W., Noh, Y.K., Park, F.C.: Transfer learning for automated optical inspection. Int. Joint Conf. Neural Netw. (IJCNN) **2017**, 2517–2524 (2017). https://doi.org/10.1109/IJCNN.2017.7966162
5. Li, J., Chen, J., Sheng, B., Li, P., Yang, P., Feng, D.D., Qi, J.: Automatic detection and classification system of domestic waste via multimodel cascaded convolutional neural network. IEEE Trans. Industr. Inf. **18**(1), 163–173 (2022). https://doi.org/10.1109/TII.2021.3085669
6. Yang, Y., Yan, L.F., Zhang, X., Han, Y., Nan, H.Y., Hu, Y.C., Hu, B., Yan, S.L., Zhang, J., Cheng, D.L., Ge, X.W., Cui, G.B., Zhao, D., Wang, W.: Glioma grading on conventional MR images: a deep learning study with transfer learning. Front. Neurosci. **12**, 804 (2018). https://doi.org/10.3389/fnins.2018.00804
7. Oztel, I., Yolcu, G. and Oz, C.: Performance comparison of transfer learning and training from scratch approaches for deep facial expression recognition. 2019 4th International Conference on Computer Science and Engineering (UBMK), 1–6 (2019). https://doi.org/10.1109/UBMK.2019.8907203.
8. Ozaltin, O., Yeniay, O.: A novel proposed CNN–SVM architecture for ECG scalograms classification. Soft. Comput. **27**(8), 4639–4658 (2023). https://doi.org/10.1007/s00500-022-07729-x
9. Oguz, C., Aydin, T. and Yaganoglu, M.: A CNN-based hybrid model to detect glaucoma disease. Multimed. Tools Appl. 1–19 (2023). https://doi.org/10.1007/s11042-023-16129-8.
10. Keerthana, D., Venugopal, V., Nath, M.K., Mishra, M.: Hybrid convolutional neural networks with SVM classifier for classification of skin cancer. Biomed. Eng. Adv. **5**, 100069 (2023). https://doi.org/10.1016/j.bea.2022.100069
11. Aslan, S., Kızıloluk, S., Sert, E.: TSA-CNN-AOA: Twitter sentiment analysis using CNN optimized via arithmetic optimization algorithm. Neural Comput. Appl. 1–18 (2023). https://doi.org/10.1007/s00521-023-08236-2.
12. Kundu, R., Chattopadhyay, S.: Deep features selection through genetic algorithm for cervical pre-cancerous cell classification. Multimed. Tools Appl. **82**(9), 13431–13452 (2023). https://doi.org/10.1007/s11042-022-13736-9
13. Li, F., Hou, J., Liu, S., Liu, X., Liu, L., Lyu, Q.: Blast furnace raw material granularity recognition model based on deep learning and multimodal fusion of 3D point cloud. Vis. Comput. **40**(10), 6939–6954 (2024). https://doi.org/10.1007/s00371-024-03449-5
14. Sun, G., Huang, D., Cheng, L., Jia, J., Xiong, C., Zhang, Y.: Efficient and lightweight framework for real-time ore image segmentation based on deep learning. Minerals **12**(5), 526 (2022). https://doi.org/10.3390/min12050526
15. Zhou, W., Wang, H., Wan, Z.: Ore image classification based on improved CNN. Comput. Electr. Eng. **99**, 107819 (2022). https://doi.org/10.1016/j.compeleceng.2022.107819
16. Chen, J., Fu, Z., Huang, J., Hu, X., Peng, T.: Boosting vision transformer for low-resolution borehole image stitching through algebraic multigrid. Vis. Comput. **38**(9), 3191–3203 (2022). https://doi.org/10.1007/s00371-022-02564-5
17. Liu, Y., Zhang, Z., Liu, X., Lei, W., Xia, X.: Deep learning based mineral image classification combined with visual attention mechanism. IEEE Access **9**, 98091–98109 (2021). https://doi.org/10.1109/ACCESS.2021.3095368
18. Olivier, L.E., Maritz, M.G., Craig, I.K.: Estimating ore particle size distribution using a deep convolutional neural network. IFAC-Pap. OnLine **53**(2), 12038–12043 (2020). https://doi.org/10.1016/j.ifacol.2020.12.740
19. Karimpouli, S., Tahmasebi, P.: Segmentation of digital rock images using deep convolutional autoencoder networks. Comput. Geosci. **126**, 142–150 (2019). https://doi.org/10.1016/j.cageo.2019.02.003
20. Singh, V., Singh, T.N., Singh, V.: Image processing applications for customized mining and ore classification. Arab. J. Geosci. **4**, 1163–1171 (2011). https://doi.org/10.1007/s12517-010-0125-2
21. Rawat, W., Wang, Z.: Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput. **29**(9), 2352–2449 (2017). https://doi.org/10.1162/NECO_a_00990
22. Sagi, O., Rokach, L.: Ensemble learning: a survey. Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. **8**(4), e1249 (2018). https://doi.org/10.1002/widm.1249

23. Amin, S.U., Alsulaiman, M., Muhammad, G., Mekhtiche, M.A., Shamim Hossain, M.: Deep learning for EEG motor imagery classification based on multi-layer CNNs feature fusion. Fut. Gen. Comput. Syst. **101**, 542–554 (2019). https://doi.org/10.1016/j.future.2019.06.027
24. Chen, Z., Zhu, M., Chen, S., Lu, L., Tang, H., Hu, X., Ji, C.: Discriminative feature fusion for RGB-D salient object detection. Comput. Electr. Eng. **106**, 108579 (2023). https://doi.org/10.1016/j.compeleceng.2023.108579
25. Karaboga, D.: Artificial bee colony algorithm. Scholarpedia **5**(3), 6915 (2010)
26. Holland, J.H.: Genetic algorithms. Sci. Am. **267**(1), 66–73 (1992)
27. Yang, X.-S., Karamanoglu, M., He, X.: Flower pollination algorithm: a novel approach for multiobjective optimization. Eng. Optim. **46**(9), 1222–1237 (2014). https://doi.org/10.1080/0305215X.2013.832237
28. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, C3P Report, 826, 37 (1989)
29. Kennedy, J., Eberhart, R.: Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, 1942–1948 vol.1944 (1995). https://doi.org/10.1109/ICNN.1995.488968.
30. Carey, C., Boucher, T., Mahadevan, S., Bartholomew, P., Dyar, M.: Machine learning tools formineral recognition and classification from Raman spectroscopy. J. Raman Spectrosc. **46**(10), 894–903 (2015). https://doi.org/10.1002/jrs.4757