DEVOTION
Journal of Research and Community Service
Volume 7, Number 10, October 2026, pp. x-xx
e-ISSN: 2797-6068
https://doi.org/10.59188/devotion.v7i8.25749

Mushroom Species Image Classification Using EfficientNetB4 with MixUp and CutMix Augmentation

Sike Laurika Olia
Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) IKMI Cirebon, Indonesia
Corresponding Author: sikelaurika625@gmail.com

Keywords:
EfficientNetB4; mushroom image classification; MixUp; CutMix, transfer learning

This is an open access article under the CC BY-SA license.

Copyright © 2026 by Author. Published by
Green Publisher

Abstract

Background: Mushroom image classification is an application of computer vision that aims to recognize mushroom categories from the visual characteristics of an image.
Objective: This study evaluates EfficientNetB4 with MixUp and CutMix augmentation for 21-class mushroom image classification.
Method: The initial dataset contained 8,857 images and 20,248 COCO annotations. Following bounding-box cropping and the removal of 107 crops that were too small, 20,141 crops were retained and split into 14,289 training, 2,461 validation, and 3,391 testing images. Four configurations were compared: a baseline, MixUp ($\alpha = 0.2$), CutMix ($\alpha = 1.0$), and a combination of MixUp and CutMix. The model used ImageNet-pretrained EfficientNetB4 with 380 × 380 inputs, five feature-extraction epochs, and 15 fine-tuning epochs.
Result: The combined configuration achieved the highest validation accuracy (88.30%), whereas CutMix achieved the highest validation macro F1-score (88.08%). CutMix was selected using the macro F1-score and achieved 86.17% testing accuracy, 84.51% macro precision, 86.60% macro recall, 85.08% macro F1-score, and 95.72% Top-3 accuracy.
Conclusion: CutMix provided slightly better class-level balance on validation, but errors remained among visually similar species, and the 90% accuracy target was not achieved. The study therefore does not demonstrate a clear benefit of MixUp or CutMix over the baseline.

INTRODUCTION

Mushroom image classification is an application of computer vision that aims to recognize mushroom categories from the visual characteristics of an image. Observable characteristics include shape, colour, texture, and the structure of parts of the fruiting body. In multi-class classification, similarities between classes can make it difficult for a model to distinguish images correctly. This makes the choice of architecture and training strategy an important aspect of developing a classification model.

The practical importance of this task stems from the diversity of fungi and the health risks of misidentification. The global number of fungal species is estimated at 2.2–3.8 million, of which only about 120,000 have been formally described (Hawksworth & Lücking, 2017), and more than 2,100 wild species are consumed as food in over 100 countries (Pérez-Moreno et al.,

2021). Because edible and toxic species can be morphologically similar, misidentification is a recurrent cause of poisoning.

In China, 10,036 mushroom poisoning outbreaks involving 38,676 illnesses and 788 deaths were reported between 2010 and 2020, making mushroom poisoning the leading cause of foodborne disease outbreaks and outbreak-associated deaths in that country (Li et al., 2021). In the United States, poison centres recorded 133,700 mushroom exposures and 52 deaths between 1999 and 2016, and misidentification of edible species was the most common cause (Brandenburg & Ward, 2018). In Indonesia, a validation of reports from 2010 to 2020 documented 76 wild-mushroom poisoning cases with 550 victims and nine deaths, and called for a national database of toxic species (Putra, 2021). Mushrooms also have considerable economic value: international trade in wild edible fungi exceeded 1.23 million tonnes in 2017 and is estimated to be worth billions of US dollars annually (Pérez-Moreno et al., 2021). Reliable image-based recognition can therefore support food-safety education, foraging decisions, and quality control in cultivation, although it cannot replace expert taxonomic verification.

Convolutional neural networks (CNNs) are widely used to learn visual features automatically, and several studies have applied them to mushrooms and other macrofungi. Ketwongsa et al. (2022) improved AlexNet to distinguish poisonous from edible mushrooms, Rahmadhani and Marpaung (2023) classified mushrooms by genus with a CNN, and Lim et al. (2024) combined machine learning and image processing to identify mushroom species in Malaysia. More recent work has added attention modules (Chakraborty et al., 2025), optimized the use of deep learning techniques (Kerta et al., 2025), and modified InceptionV3 for mushroom species (Naufal et al., 2026). For macrofungi, Kumru et al. (2025) and Sevindik et al. (2025) compared CNN architectures, fusion strategies, and explainable AI for fine-grained taxonomic classification. In the FungiCLEF challenge, deep feature extraction with gradient-boosting ensembles Desingu, et.al, (2022) and transformer-based models in an open-set scenario (Wolf & Beyerer, 2022) have been used for fine-grained fungi recognition.

Table 6 summarises the data, number of classes, models, and reported accuracies of these studies where such details are available. Reported accuracies range from 81.35% for 18 species Naufal et al., (2026) to 98.50% for five species Ketwongsa et al., (2022) and tend to be lower when more visually similar classes are involved. However, the studies used different datasets, data splits, and evaluation protocols, and none of them reported using the Kaggle/Roboflow dataset analysed here, so their values are not directly comparable with one another or with the present study. One approach to developing CNNs is transfer learning, which uses the weights of a model trained on a large-scale dataset as the starting point for learning on the target dataset. EfficientNet is a family of architectures that scales the network by balancing depth, width, and input resolution (Tan & Le, 2019; Zhao et al., 2024).

Its ImageNet-pretrained weights Deng et al., (2009) provide a generic visual representation that can be adapted to fine-grained biological images when the target dataset is limited. In this study, EfficientNetB4 is used as the backbone for mushroom image classification. In addition to the architecture, the variation of the training data affects the ability of a model to learn the characteristics of the images. MixUp creates new samples by interpolating two images and

their labels Hongyi Zhang, et.al, (2018), whereas CutMix replaces part of an image with a patch from another image and adjusts the label according to the proportion of the area Yun et al., (2019); a recent survey describes both techniques and catalogues their many variants (Jin et al., 2025).

The two techniques have different mechanisms, so they need to be compared under comparable experimental conditions. Mixing has also been combined with transfer learning for few-shot fungi classification Tam, et.al, (2025) and with explanation-driven augmentation frameworks (Z. Zhang, 2022). Nevertheless, the effect of these techniques on a crop-based, 21-class mushroom dataset with an imbalanced class distribution has not been reported under a controlled comparison.

This study therefore compares four EfficientNetB4 configurations for 21-class mushroom image classification: a baseline, MixUp, CutMix, and a combination of MixUp and CutMix. The configurations share the same dataset, data split, input size, backbone, and training schedule, so that differences in results can be attributed mainly to the augmentation treatment. The configuration with the highest validation macro F1-score is selected for a single final evaluation on the test set, and the results are analysed using overall and class-balanced metrics, the confusion matrix, and examples of misclassified images. The conclusions are limited to the dataset, data split, architecture, parameters, and experimental procedure reported in the notebook.

The novelty of this study lies in a controlled comparison of MixUp, CutMix, and their combination on a crop-based, imbalanced 21-class mushroom dataset, in which only the mixing treatment is varied and the results are assessed with class-balanced metrics, a per-class analysis, and an explicit account of statistical uncertainty. A top-1 test accuracy of 90% was set in the experimental protocol as an operational benchmark, because multi-class mushroom studies with a comparable number of classes report accuracies of around this level (Lim et al., 2024). and because lower accuracy would limit the usefulness of the model as a decision-support tool.

The benefits of the study are practical evidence on whether mixing augmentation is worth adopting for this type of dataset and a transparent baseline against which future mushroom classifiers can be compared. Its implications are that augmentation choices for fine-grained biological images should be supported by class-balanced metrics and uncertainty estimates rather than by small differences from a single run, and that any recognition system intended for practical use should complement, not replace, expert identification.

METHOD

This study used a quantitative approach with a comparative experimental design. The treatment variable was the data augmentation configuration applied to the EfficientNetB4 architecture. Four scenarios were run using the same dataset, data split, input size, and base architecture. The main difference between the scenarios lay in the use of MixUp and CutMix on the training data.

According to the notebook inspection, the dataset consists of 8,857 images, 20,248 annotations, and 21 classes. The source split contained 7,226 training, 814 validation, and 817 testing images. The images were then processed using the bounding boxes to obtain the

mushroom object regions. The number of annotations is therefore not equal to the number of images, because one image can contain more than one annotation. Table 1 shows the number of images used after cropping and quality filtering.

Table 1. Number of cropped images in the training, validation, and testing sets

Subset	Number of images	Proportion
Training	14,289	70.94%
Validation	2,461	12.22%
Testing	3,391	16.84%
Total	20,141	100%

Source: research data. The numbers refer to cropped images, not to the original images of the source dataset.

A total of 14,289 images (70.94%) were allocated for training, 2,461 images (12.22%) for validation, and 3,391 images (16.84%) for testing. The training proportion is the largest because the model requires enough examples to learn visual representations. The validation set was used to compare configurations and select the model, whereas the test set was kept for the final evaluation. This separation is important so that the test data do not influence decisions during model development. The crop subsets were not re-drawn. Each crop inherited the subset of its source image in the original export, so all crops from the same photograph remained in the same subset; the leakage audit described below confirmed that no group_id or source_id was shared between subsets.

The crop proportions therefore differ from the image proportions of the source split (81.6/9.2/9.2%) because the average number of retained annotations per image differs between subsets, at approximately 2.0, 3.0, and 4.2 crops per image in the training, validation, and testing subsets, respectively. The directory name indicates a Roboflow export (version 6). Such exports may contain augmented copies of the training images, and the larger share of source images in the training split is consistent with this possibility; however, the export settings were not recorded in the notebook, so the presence of augmented copies could neither be confirmed nor excluded.

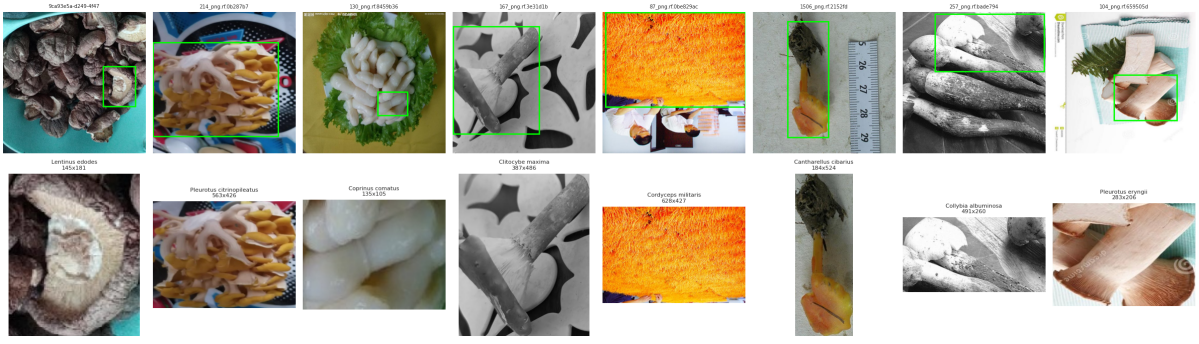

Figure 1. Examples of source images, bounding-box annotations, and cropping results
Source: research data.

Figure 1 illustrates the visual process underlying the construction of the dataset: source images are shown with their bounding-box annotations, and the object regions are then cropped into separate images. The bounding box determines the coordinates of the region that is

retained. Cropping reduces the influence of irrelevant background and makes the mushroom object more dominant in the model input. However, the quality of the crops depends on the accuracy of the annotations; a box that is too narrow can cut off important morphological parts, whereas a box that is too wide can include excessive background.



Figure 2. Gallery of cropped sample images (part 1 of 3)
Source: research data.

Figure 2 shows examples of crops from some of the classes. The gallery is useful for checking whether the mushroom objects remain clearly visible after cropping and whether variations in shape, colour, texture, and viewing angle are still represented. Intra-class variation can be seen in the differences in size and appearance of objects within the same class. This is a learning challenge because the model must recognize relatively consistent class characteristics amid visual variation.



Figure 3. Gallery of cropped sample images (part 2 of 3)
Source: research data.

Figure 3 continues the visual inspection of the other classes. The comparison between images shows that several species have similar visual characteristics, so classification decisions cannot always be determined from colour or general shape alone. A combination of texture, cap shape, stem, gill arrangement, and object context may serve as visual cues learned by the network, provided that these features are visible in the cropped region.



Figure 4. Gallery of cropped sample images (part 3 of 3)
Source: research data.

Figure 4 completes the gallery for the remaining classes. The visualization also serves as a qualitative check of the labels. Sample images are not sufficient to prove that all labels are correct, however; biological verification of every image requires examination by experts or a validated taxonomic source. The gallery is therefore treated as an inspection of image quality and annotation consistency, not as a final taxonomic validation.Preprocessing included checking the existence of files and annotations, reading image information, cropping objects based on bounding boxes, checking the crops, resizing the images to 380 × 380 pixels, and assembling the training, validation, and testing subsets.

MixUp and CutMix were applied only to the training data; the validation and test data were used without mixing augmentation to keep the evaluation consistent. The initial audit confirmed that the number of image files in each folder matched the entries in the COCO annotation files. The three source splits contained 8,857 images and 20,248 annotations, with no missing files or corrupted images recorded in the audit. Of the 20,248 crop candidates, 20,141 were usable, and 107 crops were excluded because their shortest side was below the 32-pixel threshold. Crops that are too small may lose the texture and structural details needed to distinguish species. All retained crops were then resized to 380 × 380 pixels to match the input of EfficientNetB4.

The SHA-256 check detects only byte-identical files. Near-duplicates, such as flipped, rotated, or re-encoded copies produced by export-time augmentation, are not detected by this check, and no perceptual-hash or embedding-similarity screening was performed. Because export-time augmentation is typically applied only to the training split, such copies would remain within the training subset rather than leak into the evaluation subsets, but they would reduce the effective diversity of the training data; this is acknowledged as a limitation.

The class imbalance (a ratio of 13.74 between the largest and smallest training classes) was deliberately left uncorrected by class weighting or resampling, so that the four configurations differed only in the mixing treatment. Rebalancing changes the effective class distribution seen during training, as mixing also does through its soft labels, and combining the two would have confounded the comparison (Buda et al., 2018; Johnson & Khoshgoftaar, 2019). The effect of the imbalance was instead monitored with macro-averaged and per-class metrics. EfficientNetB4 was used as the backbone with ImageNet-pretrained weights. EfficientNet uses compound scaling, which balances the depth, width, and resolution of the network through a single compound coefficient (Tan & Le, 2019; Zhao et al., 2024). The model was adapted to the multi-class task by adding a classification head on top of the backbone, consisting of batch normalization, dropout, a dense layer of 256 units with ReLU activation, dropout, and a dense layer with 21 softmax outputs. For logits z_j, the predicted probability of class j is computed with the softmax function in Equation (1), where C = 21.

$$p_j = \exp(z_j) / \Sigma_{k=1}^{C} \exp(z_k) \quad (1)$$

Training was carried out in two stages. In the feature-extraction stage, the backbone weights were frozen for five epochs so that the classification head could adapt to the target classes. In the fine-tuning stage, the top 100 layers of the backbone were unfrozen and trained for 15 epochs with a lower learning rate. Batch normalization layers in the backbone were kept in inference mode so that their statistics remained stable. Training used soft labels, and the loss for a sample with label vector y and predicted probabilities p is the categorical cross-entropy in Equation (2).

$$L = -\Sigma_{j=1}^{C} y_j \log p_j \quad (2)$$

MixUp forms a virtual training sample from two images x_i and x_j with one-hot labels y_i and y_j by linear interpolation, as in Equations (3) and (4), where the mixing coefficient λ is drawn from a Beta(α, α) distribution (Jin et al., 2025; H. Zhang et al., 2018).

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \quad (3)$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j, \lambda \sim \text{Beta}(\alpha, \alpha) \quad (4)$$

CutMix instead replaces a rectangular region of x_i with the corresponding region of x_j. With a binary mask M that equals zero inside the pasted region and one elsewhere, the mixed image and label are given by Equations (5) and (6), where $\odot$ denotes element-wise multiplication and λ is set to the proportion of the area of x_i that is retained (Jin et al., 2025; Yun et al., 2019).

$$\tilde{x} = M \odot x_i + (1 - M) \odot x_j \quad (5)$$

$$\tilde{y} = \lambda\, y_i + (1 - \lambda)\, y_j \quad (6)$$

The four scenarios are summarized in Table 2 and Figure 5. E1 serves as the baseline to determine the performance of EfficientNetB4 without image mixing. E2 isolates the effect of MixUp with $\alpha = 0.2$, whereas E3 isolates the effect of CutMix with $\alpha = 1.0$. These values follow the original studies: $\alpha = 0.2$ lies within the range of 0.1–0.4 that Zhang et al. (2018) found effective on ImageNet, and $\alpha = 1.0$ is the setting used by Yun et al. (2019). E4 combines the two approaches alternately, with a selection probability of 0.5 for each technique on the training batches. Because the backbone, input size, data split, and training schedule were kept the same, the comparison was designed so that changes in results could be attributed mainly to the augmentation configuration. Full sensitivity testing of α was not enabled in the main experimental configuration, and the experiments were not repeated with multiple seeds.

Table 2. EfficientNetB4 experimental scenarios

Code	Configuration	Augmentation parameters
E1	EfficientNetB4 baseline	No MixUp/CutMix
E2	EfficientNetB4 + MixUp	$\alpha = 0.2$
E3	EfficientNetB4 + CutMix	$\alpha = 1.0$
E4	EfficientNetB4 + MixUp/CutMix	Probability MixUp 0.5; CutMix 0.5

Source: research design.

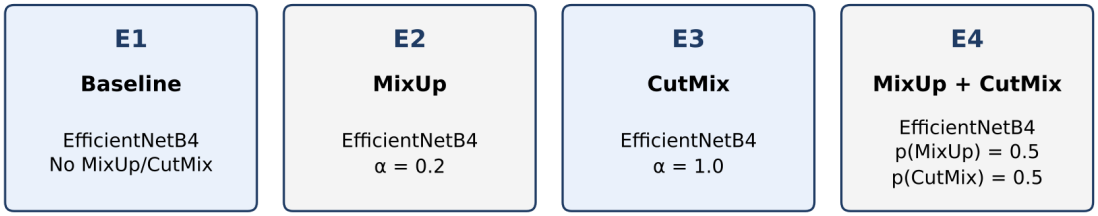

Figure 5. Configuration of the four EfficientNetB4 experimental scenarios
Source: research design.

Evaluation used accuracy, macro precision, macro recall, macro F1-score, weighted F1-score, and Top-3 accuracy. For class c, precision, recall, and F1-score are computed from the true positives (TP), false positives (FP), and false negatives (FN), as in Equations (7)–(9). Macro averaging, as in Equation (10), gives every class the same weight; this is relevant when the class distribution is imbalanced, because classes with few samples still contribute equally to the average. The weighted F1-score weights each class by its number of samples n_c out of N, as in Equation (11), and is therefore more strongly influenced by classes with large support.

$$\text{Precision}_c = TP_c / (TP_c + FP_c) \quad (7)$$

$$\text{Recall}_c = TP_c / (TP_c + FN_c) \quad (8)$$

$$F1_c = 2 \times \text{Precision}_c \times \text{Recall}_c / (\text{Precision}_c + \text{Recall}_c) \quad (9)$$

$$\text{Macro-F1} = (1/C) \sum_c F1_c \quad (10)$$

$$\text{Weighted F1} = \Sigma_c \, (n_c / N) \, F1_c \quad (11)$$

Top-3 accuracy measures whether the true label is among the three predictions with the highest scores. It provides additional information about how close the candidate predictions are, but it does not replace top-1 accuracy as the measure of the main predicted class. The configuration for the final test was selected based on the macro F1-score on the validation data. Because each configuration was trained only once, statistical uncertainty was quantified from the evaluation data rather than from repeated training. For each accuracy value, a 95% Wilson score confidence interval was computed from the number of correctly classified images (Wilson, 1927).

For the selected model, 95% percentile bootstrap confidence intervals for test accuracy and macro F1-score were obtained from 2,000 resamples of the 3,391 test predictions (Efron & Tibshirani, 1993), and per-class precision, recall, F1-score, and support were derived from the test confusion matrix. Paired tests such as McNemar's test (Dietterich, 1998) require the image-level validation predictions of all four configurations, which were not retained in the analysed notebook outputs; these tests were therefore not performed.

RESULTS AND DISCUSSION

Comparison of Experiments on the Validation Set

The results of the four configurations on the validation data are presented in Table 3. All experiments used the same data split and base architecture, so the differences in results relate mainly to the augmentation treatment. E4 recorded the highest accuracy of 88.30%, whereas E3 recorded the highest macro recall of 89.58% and the highest macro F1-score of 88.08%. The differences between scenarios are relatively small, so the interpretation is not based on a single metric.

Table 3. Comparison of the four experiments on the validation data

Experiment	Accuracy	Precision macro	Recall macro	F1-score macro	F1 weighted	Top-3 accuracy
E1 – Baseline	88.13%	87.65%	88.84%	87.89%	88.06%	96.46%
E2 – MixUp	88.22%	87.54%	88.59%	87.76%	88.15%	96.71%
E3 – CutMix	88.13%	87.40%	89.58%	88.08%	88.14%	96.42%
E4 – MixUp + CutMix	88.30%	87.64%	89.20%	88.05%	88.29%	96.34%

Source: research data.

Table 3 shows that an increase in accuracy due to augmentation is not always followed by an increase in macro F1-score. E2 produced an accuracy of 88.22%, slightly above the baseline of 88.13%, but its macro F1-score decreased from 87.89% to 87.76%. E3 maintained an accuracy of 88.13% and increased the macro F1-score to 88.08%, a gain of 0.19 percentage points over the baseline. E4 produced the highest accuracy (88.30%) and the highest weighted F1 (88.29%), but its macro F1-score of 88.05% was still slightly below that of E3. These differences indicate that E4 was better in overall correctness and in performance weighted by

the number of samples, whereas E3 was slightly more balanced when every class is treated equally.

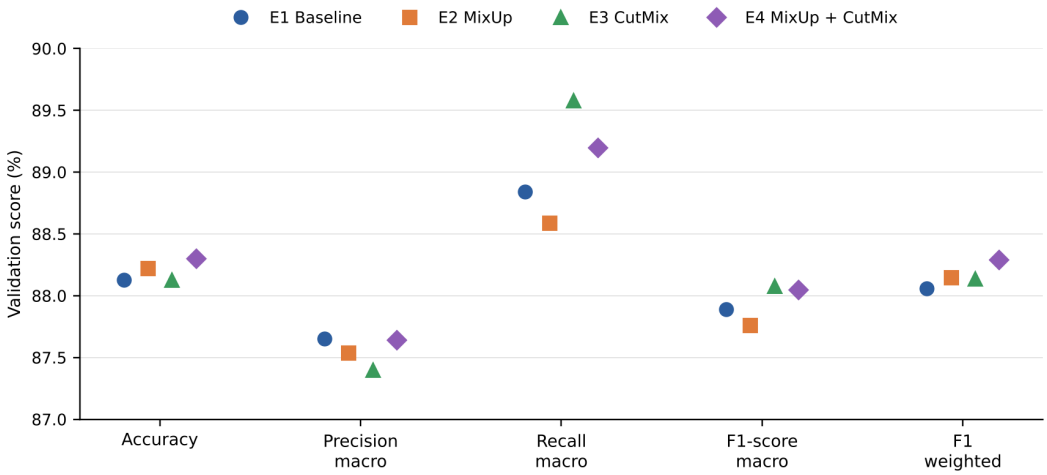

Figure 6. Comparison of validation metrics for the four experiments
Source: research data.

Figure 6 compares the validation metrics of the four configurations. E4 had the highest accuracy, whereas E3 had the highest macro F1-score and macro recall, which confirms that one model does not have to be superior on all metrics. The Top-3 accuracy of all configurations was above 96%, meaning that the actual label was often among the top three candidates. All differences between scenarios were within one percentage point on every metric; without repetition with several random seeds and statistical testing, these differences cannot yet be described as significant.

The 95% Wilson confidence intervals of validation accuracy were 86.80–89.35% for E1, 86.88–89.43% for E2, 86.80–89.35% for E3, and 86.97–89.51% for E4. Each interval spans about 2.6 percentage points and the four intervals overlap almost completely, whereas the largest difference between configurations was 0.17 percentage points in accuracy and 0.32 percentage points in macro F1-score. The ranking of the configurations is therefore not statistically supported, and the four configurations should be regarded as performing equivalently on the validation data.

Model selection was based on the validation macro F1-score rather than on accuracy alone. E3 was selected because it obtained the highest macro F1-score of 88.08%. This decision is consistent with the aim of multi-class classification, which needs to consider performance across all classes, including classes with relatively few samples. E4 remained the configuration with the highest validation accuracy, so the study does not claim that CutMix is superior on all metrics; the advantage of E3 is limited to the macro F1-score criterion that was set in advance. The selection of E3 should therefore be read as the outcome of a predefined decision rule, not as evidence that CutMix is the best configuration.

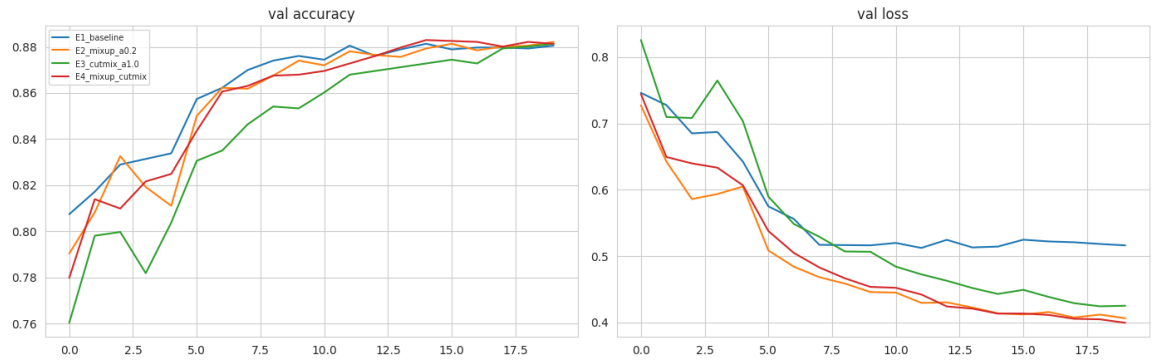

Figure 7. Validation accuracy and validation loss per epoch of the four experiments recorded in the notebook (epochs indexed from 0)

Source: research data.

Figure 7 shows the validation history recorded in the notebook. The accuracy and loss curves per epoch can be used to check the stability of the learning process. The validation loss of the baseline levelled off at around 0.52 after the feature-extraction stage, whereas the three mixing configurations continued to decrease to around 0.40–0.43. Part of this difference may reflect the softer targets produced by mixing, so loss values should be compared with caution. Interpretation of the curves needs to be made together with the final values and the train–validation gap; a single gap value is not sufficient to conclude overfitting.

Final Evaluation on the Test Set

Model E3, selected on the basis of the validation macro F1-score, was then evaluated once on the test data containing 3,391 images. The test data were not used to select α or for tuning. The results of the final test are shown in Table 4. This separate evaluation provides an estimate of the generalization ability of the model on images that were not used during training or configuration selection. The test results of E1, E2, and E4 were not computed, because the protocol reserved the test set for a single evaluation of the configuration chosen on the validation data; this limits the transparency of the comparison and is noted as a limitation.

Table 4. Evaluation results of model E3 (CutMix) on the test data

Metric	Result
Accuracy	86.17%
Precision macro	84.51%
Recall macro	86.60%
F1-score macro	85.08%
F1-score weighted	86.35%
Top-3 accuracy	95.72%
Number of test images	3,391

Source: research data.

Table 4 shows that E3 achieved an accuracy of 86.17%, meaning that about 86 out of every 100 test images were predicted correctly at top-1. The macro precision of 84.51% indicates the average correctness of predictions across classes, whereas the macro recall of 86.60% indicates the average proportion of actual objects found by the model across all classes. The macro F1-score of 85.08% summarizes the balance between precision and recall. The

weighted F1 of 86.35% was higher than the macro F1, indicating that classes with more samples contributed relatively better performance than the equal-weight average across classes. The Top-3 accuracy of 95.72% shows that the correct label was among the three top predictions for most images, although the top-1 prediction was still wrong in some cases. The 95% Wilson interval for test accuracy was 84.97–87.29%, and the bootstrap intervals were 85.05–87.26% for accuracy and 83.74–86.36% for macro F1-score; the upper bound for accuracy therefore remains below 90%.

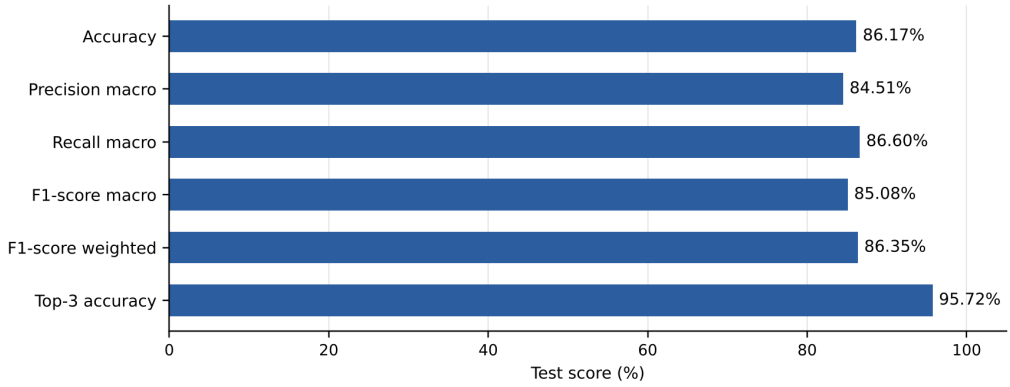

Figure 8. Evaluation metrics of model E3 on the test data
Source: research data.

Figure 8 summarizes the final evaluation metrics of E3 on the test set. The top-1 accuracy of 86.17% is lower than the Top-3 accuracy of 95.72%. The difference of 9.55 percentage points shows that the model fairly often placed the actual class among the top three candidates but did not always assign the highest probability to the correct class. Top-3 accuracy is useful as additional information about the model's candidates, but it cannot replace top-1 accuracy for assessing the main classification.

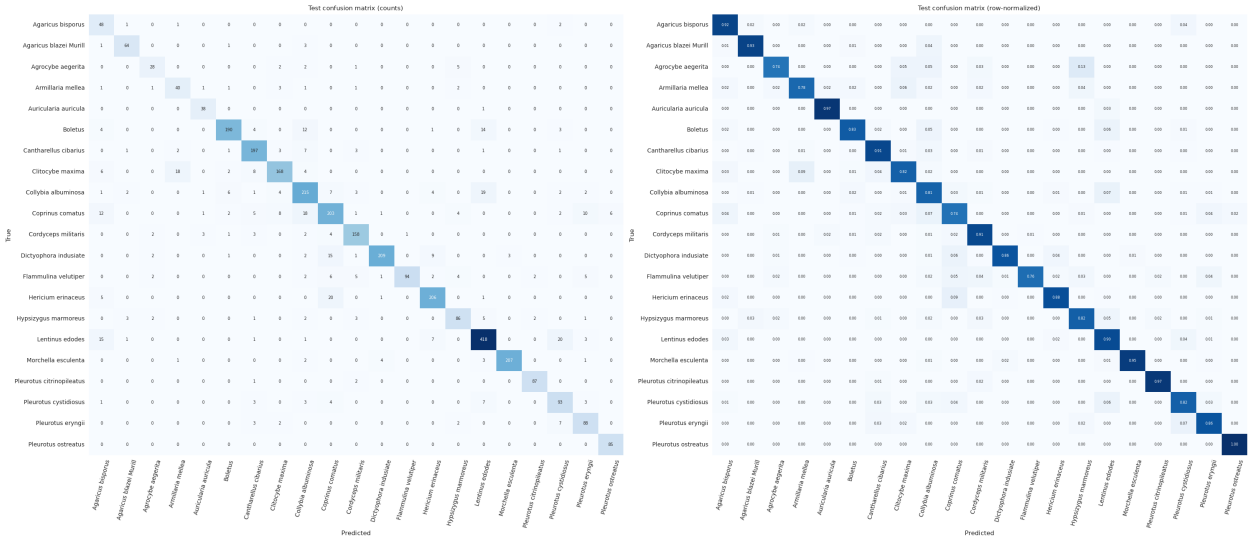

Figure 9. Test confusion matrix for 21 classes (counts and row-normalized)
Source: research data.

Figure 9 presents the confusion matrix in two forms. The left panel shows the number of predictions, whereas the right panel normalizes each row by the number of actual images in

that class. The vertical axis represents the true class and the horizontal axis the predicted class. Diagonal cells show correct predictions, and off-diagonal cells show pairs of classes that were confused. Row normalization helps to compare recall between classes even though the number of samples per class differs. The dominant misclassification patterns in the notebook included Lentinus edodes predicted as Pleurotus cystidiosus (20 images), Hericium erinaceus predicted as Coprinus comatus (20 images), and Collybia albuminosa predicted as Lentinus edodes (19 images).

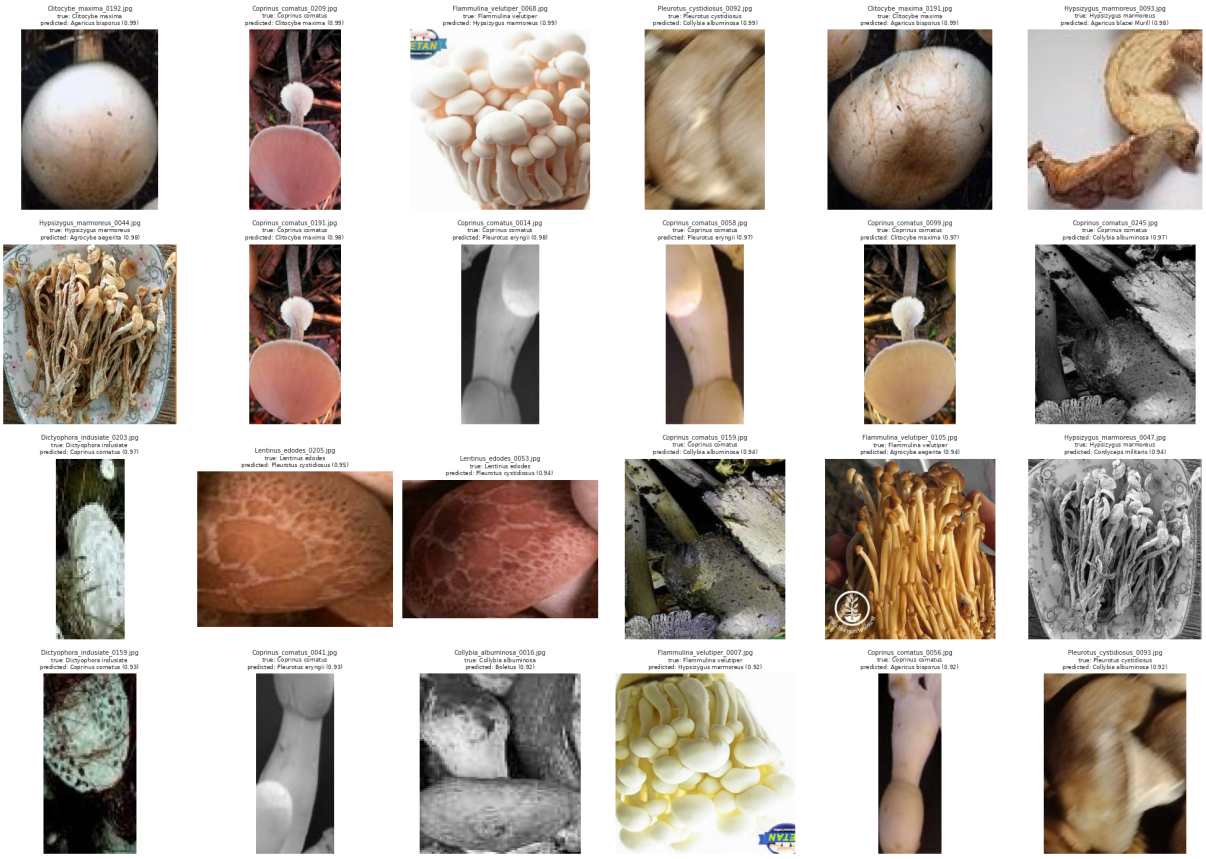

Figure 10. Examples of misclassified test images with the true label, predicted label, and confidence
Source: research data.

Figure 10 shows examples of misclassified test images together with their true labels, predicted labels, and confidence. This visualization helps to trace whether errors are related to similarities in shape, colour, or texture, crop quality, viewing angle, or background that is still included. High confidence on wrong predictions deserves attention because it shows that the model can be wrong with great certainty; all 24 examples shown were predicted with a confidence of at least 0.92. Several examples are blurred, low-resolution, or show only part of the fruiting body, such as a stem or a section of the cap. The specific cause of each error must nevertheless be examined on the image concerned and cannot be inferred from the class names alone.

Discussion

The comparison on the validation data shows that the augmentation techniques produced different changes in performance. MixUp in E2 produced an accuracy of 88.22%, slightly

higher than the baseline of 88.13%, but its macro F1-score of 87.76% was lower than the baseline value of 87.89%. CutMix in E3 produced the highest macro F1-score of 88.08%, whereas the combination of MixUp and CutMix in E4 produced the highest accuracy of 88.30%. A plausible reason why CutMix favoured class-level balance is that it keeps local regions of each image intact, so that discriminative local textures such as gill patterns or cap surfaces remain visible in the mixed samples, whereas MixUp blends whole images and may blur such details (Jin et al., 2025). Because these differences are smaller than the sampling uncertainty of the validation metrics, this mechanism should be regarded as a hypothesis rather than as the explanation of a demonstrated effect.

These differences show that the effect of augmentation needs to be examined through several metrics. Accuracy describes overall correctness, whereas the macro F1-score describes the balance between precision and recall across classes. Based on the macro F1-score criterion, CutMix was the selected configuration on the validation set. In the final test, the accuracy of E3 (86.17%) was lower than its validation accuracy (88.13%), a decrease of 1.96 percentage points. This shows that performance on data not used in model selection was lower than on the validation data, and the results do not support a claim that the 90% accuracy target has been achieved. The 90% benchmark was defined in the experimental protocol on the basis of accuracies reported for multi-class mushroom datasets with a similar number of classes (Lim et al., 2024), and even the upper bound of the 95% confidence interval for test accuracy (87.29%) lies below it.

The confusion pairs observed in the test set involve species that can resemble one another in cropped images, particularly when only part of the fruiting body is visible. Studies on fine-grained macrofungi classification have likewise reported that visually similar taxa remain the main source of errors and that explainable AI can help to examine whether models attend to diagnostically relevant structures (Kumru et al., 2025; Sevindik et al., 2025). Adding attention mechanisms Chakraborty et al., (2025) or interpretability analysis could therefore be useful directions for reducing and understanding these confusions.

Limitations

This study has several limitations. First, the copy of the notebook that was analysed stores the dataset in a directory named “mushroom.v6i.coco”, but it does not explicitly record the URL of the Kaggle page or the identity of the dataset owner; the cited Kaggle page therefore needs to be matched with the version that was actually downloaded. Second, the number of crops is imbalanced across classes, with a ratio of 13.74 between the largest and smallest classes in the training split. This imbalance can affect learning and means that overall accuracy does not fully describe the performance of each class.

Third, some source images have more than one annotation, and the quality of the bounding boxes may vary. Although the audit recorded the exclusion of 107 crops that were too small and found no leakage between splits based on group_id, source_id, and SHA-256, it does not replace manual verification of all annotations. In addition, SHA-256 hashing does not detect near-duplicate images, and it could not be determined whether the Roboflow export contained augmented copies of the training images. Fourth, the main experiment used only one α value for MixUp (0.2) and one for CutMix (1.0) because FULL_SENSITIVITY was set to False, so

the study cannot conclude that these values are optimal in general. Fifth, each scenario was reported from a single training run with seed 42, and no multi-seed repetitions are available to measure the variation and significance of the differences in performance.

Sixth, the test accuracy of 86.17% is still below the 90% target, so the model has not yet met the performance target that was set. Seventh, the confusion matrix and misclassification examples show error patterns, but no evaluation by a mycologist has been carried out to confirm whether certain labels or morphological characteristics cause the errors. Finally, the model was evaluated only on the dataset used in this study; generalization to field images and to different cameras, lighting, backgrounds, and species populations has not been tested. Paired significance tests such as McNemar's test could not be applied because image-level predictions of all configurations were not retained, and only the selected configuration was evaluated on the test set.

CONCLUSION

This study compared four EfficientNetB4 configurations (baseline, MixUp, CutMix, and MixUp/CutMix) for 21-class mushroom image classification. On the validation data, all configurations performed within one percentage point of each other (accuracy 88.13–88.30%; macro F1-score 87.76–88.08%), and these differences were not tested for significance across repeated runs and fell within the sampling uncertainty.

The study therefore does not demonstrate a clear benefit of MixUp or CutMix over the baseline. CutMix, selected by the predefined macro F1-score criterion, achieved 86.17% accuracy, 85.08% macro F1-score, and 95.72% Top-3 accuracy on the test data, below the 90% benchmark. These findings imply that augmentation choices for fine-grained mushroom datasets should be validated with repeated runs and class-level evaluation, while label verification and additional data for confusable species are likely to yield larger gains.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. The author did not receive any financial support for the conduct of this research or for the publication of this article.

CONFLICT OF INTEREST

The author declares no conflict of interest. This research was conducted in the absence of any commercial or financial relationship that could be construed as a potential conflict of interest. No external party had any role in the study design, data collection, analysis, interpretation of the results, writing of the manuscript, or the decision to submit it for publication.

AUTHOR CONTRIBUTION

Sike Laurika Olia: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Project administration, Writing – original draft, and Writing – review & editing. The author has read and approved the final version of the manuscript.

GENERATIVE ARTIFICIAL INTELLIGENCE USAGE STATEMENT

The author used generative artificial intelligence (AI) tools during the preparation of this manuscript solely to improve language clarity and grammatical structure and to assist with the formatting of reference citations. Generative AI was not used to generate or modify research data, to design or run the experiments, to perform the statistical analyses, to produce the figures, or to interpret the results. All AI-assisted output was reviewed and edited by the author. This use complied with applicable ethical guidelines on the use and disclosure of generative AI in scientific research. The author takes full responsibility for the integrity, accuracy, and final content of the manuscript.

REFERENCES

Brandenburg, W. E., & Ward, K. J. (2018). Mushroom poisoning epidemiology in the United States. *Mycologia*, *110*(4), 637–641. https://doi.org/10.1080/00275514.2018.1479561

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, *106*, 249–259. https://doi.org/10.1016/j.neunet.2018.07.011

Chakraborty, B., Mukherjee, R., & Mandal, S. (2025). Convolutional block attention-based deep neural network for mushroom classification. *Discover Artificial Intelligence*, *5*(1), 252. https://doi.org/10.1007/s44163-025-00488-2

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Desingu, K., Bhaskar, A., Palaniappan, M., Chodisetty, E. A., & Bharathi, H. (2022). Classification of fungi species: A deep learning based image feature extraction and gradient boosting ensemble approach. *CLEF 2022: Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings*.

Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, *10*(7), 1895–1923. https://doi.org/10.1162/089976698300017197

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman & Hall/CRC. https://doi.org/10.1201/9780429246593

Hawksworth, D. L., & Lücking, R. (2017). Fungal diversity revisited: 2.2 to 3.8 million species. *Microbiology Spectrum*, *5*(4), FUNK-0052-2016. https://doi.org/10.1128/microbiolspec.FUNK-0052-2016

Jin, X., Zhu, H., Li, S., Wang, Z., Liu, Z., Tian, J., Yu, C., Qin, H., & Li, S. Z. (2025). A survey on Mixup augmentations and beyond. In *arXiv*. https://doi.org/10.48550/arXiv.2409.05202

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*(1), 27. https://doi.org/10.1186/s40537-019-0192-5

Kerta, J. M., Rangkuti, A. H., Lau, S. L., Kurniawan, A., Gabriela, M., & Tandianto, A. (2025). Improving accuracy through optimal use of deep learning techniques for mushroom image classification. *International Journal on Informatics Visualization*, *9*(3), 1298–1309.

Ketwongsa, W., Boonlue, S., & Kokaew, U. (2022). A new deep learning model for the

classification of poisonous and edible mushrooms based on improved AlexNet convolutional neural network. *Applied Sciences*, *12*(7), 3409. https://doi.org/10.3390/app12073409

Kumru, E., Ekinci, F., Aydoğan, A., Güzel, M. S., Sevindik, M., & Akata, I. (2025). Explainable convolutional neural network architectures for high-performance taxonomic classification of gasteroid macrofungi. *Scientific Reports*, *15*, 40196. https://doi.org/10.1038/s41598-025-23974-0

Li, W., Pires, S. M., Liu, Z., Liang, J., Wang, Y., Chen, W., Liu, C., Liu, J., Han, H., Fu, P., & Guo, Y. (2021). Mushroom poisoning outbreaks—China, 2010–2020. *China CDC Weekly*, *3*(24), 518–522. https://doi.org/10.46234/ccdcw2021.134

Lim, J. Y., Wee, Y. Y., & Wee, K. K. (2024). Machine learning and image processing-based system for identifying mushrooms species in Malaysia. *Applied Sciences*, *14*(15), 6794. https://doi.org/10.3390/app14156794

Naufal, M. K., Sari, C. A., Rachmawanto, E. H., & Iqtait, M. (2026). Performance enhancement of mushroom species classification via modified InceptionV3. *Jurnal Masyarakat Informatika*, *17*(1), 54–69. https://doi.org/10.14710/jmasif.17.1.73005

Pérez-Moreno, J., Mortimer, P. E., Xu, J., Karunarathna, S. C., & Li, H. (2021). Global perspectives on the ecological, cultural and socioeconomic relevance of wild edible fungi. *Studies in Fungi*, *6*(1), 408–424. https://doi.org/10.5943/sif/6/1/31

Putra, I. P. (2021). Kasus-kasus keracunan jamur liar di Indonesia. *Jurnal Ekologi Kesehatan*, *20*(3), 215–230.

Rahmadhani, U. S., & Marpaung, N. L. (2023). Klasifikasi jamur berdasarkan genus dengan menggunakan metode CNN. *Jurnal Informatika: Jurnal Pengembangan IT*, *8*(2), 169–173. https://doi.org/10.30591/jpit.v8i2.5229

Sevindik, M., Korkmaz, A. F., Ekinci, F., Kumru, E., Altındal, Ö. B., Aydın, A., Güzel, M. S., & Akata, I. (2025). A comparative analysis of CNN architectures, fusion strategies, and explainable AI for fine-grained macrofungi classification. *Biology*, *14*(12), 1733. https://doi.org/10.3390/biology14121733

Tam, J. K., Gustineli, M., & Miyaguchi, A. (2025). Transfer learning and Mixup for fine-grained few-shot fungi classification. *CLEF 2025 Working Notes*.

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th International Conference on Machine Learning*, *97*, 6105–6114.

Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, *22*(158), 209–212. https://doi.org/10.1080/01621459.1927.10502953

Wolf, S., & Beyerer, J. (2022). Transformer-based fine-grained fungi classification in an open-set scenario. *CLEF 2022: Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings*.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). CutMix: Regularization strategy to train strong classifiers with localizable features. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6023–6032.

https://doi.org/10.1109/ICCV.2019.00612

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*.

Zhang, Z. (2022). *Explanation-driven frameworks for data augmentation and supervised contrastive learning in image classification* [Master's thesis]. University of Toronto.

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, *57*, 99. https://doi.org/10.1007/s10462-024-10721-6