Classifying pests in crop images using deep learning

Gabriel Sávio de Lima Mota, Leandro H. F. P. Silva,
Larissa Ferreira Rodrigues Moreira and João Fernando Mari
Instituto de Ciências Exatas e Tecnológicas
Universidade Federal de Viçosa - UFV
Rio Paranaíba, MG, Brazil
Email: {gabriel.mota, leandro.furtado, larissa.f.rodrigues, joaof.mari}@ufv.br

***Abstract*—Pest control is essential for agricultural success, and rapid and accurate pest identification through computer vision and machine learning enables effective pest management. This paper proposes an approach to evaluate nine customizations of the IP102 dataset. Considering the extensive range of sub-datasets, a comparative analysis was conducted between different deep learning models, including ResNet and AlexNet Convolutional Neural Networks (CNNs), and Vision Transform (ViT). We carried out tests considering training from scratch and fine-tuning. Our experimental results demonstrate that ViT outperforms CNN models for the problem investigated and benefits significantly from data augmentation strategies. Our study provides valuable insights for efficient pest classification, paving the way for future research and advancements in precision agriculture.**

***Index Terms*—computer vision, agriculture, pest classification, deep learning, data augmentation, cutmix**

I. Introduction

Agriculture is one of the most critical sectors in the global economy, and several countries have the largest share of their wealth in agriculture. Brazil is a significant world agricultural producer [1]. The South American country is recognized worldwide for its favorable geographical location, climate, and considerable government investment in encouraging agriculture. According to official data from the Brazilian Institute of Geography and Statistics (IBGE), Brazil's Gross Domestic Product grew by 1.9 % in the first quarter of 2023 compared to the previous quarter. Most of this growth is due to the agricultural sector, at a rate of 21.6%. According to the same institute, several crops show growth trends for the year, such as soybean (24.7%), corn (8.8%), tobacco (3%), and cassava (2.1%) [2].

Pest control is an important challenge in agricultural production. Late actions to control pest infestation may lead to several losses in production or even a reduction in product quality. Because of the large number of pests that may harm production, producers may have difficulties identifying rapidly and precisely the type of infestation that affects this culture. In this context, computer vision and machine learning tools can provide valuable solutions for improving the precision and velocity of pest detection in the field [3].

This work aims to analyze deep learning models and training strategies to solve the problem of classifying pests, more specifically insect pests, in agricultural images. We tested three deep learning classification models, two Convolutional Neural Network models, AlexNet and ResNet-50, and one attention-based model, Vision Transformer. The models are trained using combinations of data augmentation strategies, including CutMix data augmentation, and the performance was compared considering different sets of classes of the IP102 dataset.

Our results provide valuable insights into the capacity of deep learning models to identify and classify crop pests using digital images. The results and conclusions may be used to support future applications for the rapid identification of insects in plantations using handheld imaging devices, such as smartphones, imaging devices mounted in unmanned aerial vehicles (UAVs), or autonomous robots [4]–[6]. In addition, computer vision techniques for detecting and classifying pests can lead to more efficient use of pesticides, faster and more adequate responses to attacks by specific insects, and reduced crop damage and environmental impact.

This paper is organized as follows. After this introduction, we analyzed some related work to present a big picture of the state-of-the-art pest classification through computer vision in Section II. In Section III, we present the proposed method to analyze strategies to train and test deep learning models to classify pests in agricultural images. The results obtained by running our experiments over the IP102 dataset are presented in Section IV, and we conclude the paper in Section V.

II. Related Work

Besides automatic systems to detect and classify pests and insects in crops, recent advances in machine learning and computer vision enable potent tools to improve these systems. This section provides an overview of the current state-of-the-art works using computer vision to solve this problem.

Ren et al. [7] proposed the Feature Reuse Residual Network (RF-ResNet) for classifying pests in images. The network was evaluated with the IP102 dataset. Ung et al. [8] applied multiple models based on CNN and attention mechanisms. The models were also evaluated over the IP102 dataset, achieving 74.13% accuracy.

Ullah et al. [9] developed a deep learning model to classify pests in crops named DeepPestNet. The model consists of eleven trainable layers, achieving 100% accuracy in 10 classes of pests. Li et al. [10] proposed the SAFFPest that

implements a deformable convolution to detect pests in rice plants. Nanni et al. [11] developed approaches based on CNNs for pest identification. The methods are inspired by different architectures (e.g., EfficientNet B0, ResNet-50, GoogLeNet, ShuffleNet, MobileNet V2, and DenseNet-201), with different variations of the Adam optimizer, with the best performing one achieving an accuracy of 95.52% on the Deng dataset, 74.11% on the IP102 dataset, and 99.81% on Xie2.

An et al. [12] presented an approach to the problem of insect recognition based on the fusion of complementary features from multiple perspectives. Using the ResNet and Vision Transformer models, the study showed considerable ability to identify subtle differences in insect species, achieving significant results in the IP102 dataset. Zhang et al. [13] used the YOLO5 combined with a lightweight module inspired by MobileNetV3, named C3M.

Zheng et al. [5] created the Pest Classification Network (PC-Net), an approach that utilizes the EfficientNet V2 embedded attention mechanism to identify various insect pests, particularly in their larval stage. Guo et al. [14] proposed a multi-label classification method to address the class imbalance issue on the IP102 dataset and evaluated the Swin Transformer.

Although the previously mentioned works focused on pest classification using computer vision, most of them have not dealt adequately with the problem of class imbalance and have not investigated the potential of different data augmentation strategies, including CutMix. To fill this gap, our paper proposes an approach to select the more relevant classes and evaluate different deep learning models.

III. Material and methods

A. Dataset

We used the IP102 dataset [15] to perform experiments and evaluate our training strategies. The IP102[1] contains 75,222 images organized in 102 classes, and it is provided with a set of images designed for classification tasks and another one for detection tasks. In this work, we used the classification task dataset. The classes of the IP102 dataset are related to insect pests in diverse agricultural crops. These classes are organized hierarchically, and the dataset is divided into two large groups in accordance with the crop classification: field crop (FC) and economic crop (EC). Each crop group contains five and three crops, named super-classes, respectively. Each super-class contains a number of classes representing pests that affect that type of crop. Also, the IP102 is provided with non-overlapping training, validation, and test sets, with proportions 60%, 10%, and 30%, respectively.

The IP102 dataset presents challenging features, including a very large number of highly imbalanced classes. The classes contain images from very different sources, such as photos and drawings, some of them with watermarks. Also, the images present a high variety of poses and zoom levels, and some insect species are imaged in different growing stages, such as eggs, larvae, pupas, and adult insects. These features make the IP102 a challenging and, at the same time, valuable resource to study the capacity of building machine learning-based models to automatically classify these menaces.

[1]https://github.com/xpwu95/IP102

B. Architectures

For this work, we trained three deep learning architectures, two CNNs, AlexNet and ResNet-50, and one attention-based architecture, Vision Transformer (ViT).

AlexNet [16] stands as one of the pioneering deep learning architectures that gained widespread recognition. This neural network is characterized by its composition of multiple convolutional layers followed by fully connected layers. It achieved victory in the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Due to its historical significance and simplicity when compared with newer deep learning models, AlexNet has been selected for this study.

Residual Network (ResNet) [17] is a CNN architecture distinguished by its innovative concept of residual connections. These connections enable the gradients to propagate effectively through numerous layers without diminishing significantly. This approach addresses the gradient vanishing problem encountered in very deep networks.

Vision Transformer (ViT) [18] is not a CNN model but a transformer architecture applied to image classification. ViT divides the input image into fixed patches, linearizes them, and processes them sequentially, resembling the operation of transformers commonly used in natural language processing tasks. ViT was incorporated into the project due to its promising ability to capture long-range dependencies in images and its widespread popularity and superior performance across various computer vision benchmarks.

C. Data augmentation

Data augmentation is a regularization technique that consists of artificially augmenting the number of images available to train a model based on random transformations and perturbations over the original training images. Data augmentation can improve the capability of generalization of the models, and it is useful to deal with small datasets [19].

Besides more traditional data augmentation transformations, we employed and evaluated the impact of the CutMix data augmentation [20]. CutMix consists of combining information from two different images. During the training, patches of images are cut and pasted into other images in the same batch. The labels are also combined proportionally to the image regions. For example, if a patch is extracted from 25% of a source image and pasted in a destination image, the label of the image presented to the input layer is 75% of the label of destine images and 25% of the label of the source image. Figure 1 illustrates the CutMix applied over a batch of four images.

D. Experiment design

Before starting the experimental setup, we critically analyzed the dataset and observed that the dataset was heavily unbalanced. We avoid the use of classes with a small number

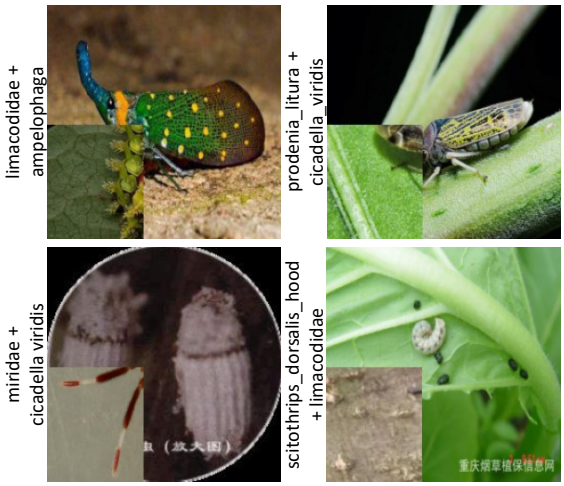

Fig. 1. Examples of the CutMix data-augmentation strategy applied over a batch with size 4.

of samples. As the dataset is hierarchically organized, we selected only the ten and twenty classes with more images in each crop group, EC and FC, generating the datasets EC-10, FC-10, EC-20, and FC-20. We also generated two datasets with the ten and twenty classes with more images considering the whole dataset, named Full-10 and Full-20. Figure 2(a) illustrates the selection of the ten and twenty classes for Full-10 and Full-20 through a histogram of the number of images in each class. Figures 2(c) illustrate the classes selected to be part of the datasets EC-10 and EC-20, and Figure 2(b) illustrates the classes selected to be part of datasets EC-10 and FC-20.

We fine-tuned three deep-learning classification models, AlexNet, ResNet-50, and ViT, using the Stochastic Gradient Descending optimizer (SGD) with a learning rate of 0.0001 and momentum of 0.9. During the training, the learning rate was decreased by a strategy named reduce learning rate on plateau, where we decreased the current learning rate by a factor of 0.1 when the validation loss did not improve for 10 epochs. The training process is stopped early if the validation loss does not decrease for 20 consecutive epochs or the training reaches a maximum of 200 epochs. We used a batch size of 256 for AlexNet, 64 for Resnet-50, and 32 for ViT. The batch sizes for ResNet-50 and ViT are smaller because the numbers of trainable parameters are higher than AlexNet, and we used the maximum batch size that the models fit in the GPU memory. The pre-trained models were obtained from the torchvision library, and we kept all model layers unfroze, i.e., all network parameters were fine-tuned during the training.

Each model was trained considering three data augmentation strategies: a) no data augmentation, b) data augmentation without CutMix, and c) data augmentation strategies with the CutMix. For the training strategy without data augmentation, the images were randomly cropped and resized to 224 $\times$ 224 pixels (random resized crop transformation), followed by normalization considering the mean and standard deviation of the ImageNet dataset, because the models that we fine-tuned were pre-trained with the ImageNet dataset. This image transformation strategy was used for the validation and test sets across all strategies and experiments. When we trained the models with data augmentation, images from the train set were submitted to a random resized crop, followed by a random horizontal flip, random vertical flip, random rotation (30°), random sharpness adjustment, random auto contrast and normalization using the mean and standard deviation of the ImageNet dataset. Finally, when we trained their models with CutMix, we applied all transformations considered in the previous strategy plus the CutMix data augmentation strategy.

E. Model evaluation

We evaluate the trained models over the validation and test sets using accuracy, precision, recall, and f1-score metrics. By comparing the metrics obtained with the validation and testing sets, it is possible to evaluate the capability of the models to extrapolate the knowledge learned to unseen samples.

Accuracy, precision, recall and f1-score are defined by Equations 1, 2, 3, and 4:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\text{-}score = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{4}$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative samples, respectively.

F. Computational resources

Experiments were executed on three PC computers with i5 3.0 GHz processors and 32 GB of RAM. Two of them are equipped with GPUs NVIDIA 1080 Ti with 11 GB of memory and the other one with an NVIDIA Titan Xp with 12 GB of RAM. The development environment is based on Python 3.9 programming language, PyTorch 2.0.1, with CUDA 11.7. The libraries torchvision 0.15.2, scikit-learn 1.3, and Matplotlib 3.7.2 were also used.

IV. Results and Discussion

Table I presents the results obtained over the validation set for the AlexNet, ResNet-50, and ViT models. The results are computed regarding accuracy, precision, recall, and f1-score. The tables also show the number of epochs each model was trained (column Epochs), considering the training was stopped according to an early stopping strategy based on the validation loss, as described in Section III-D.

Furthermore, we used the trained models to predict the classes of images from a new, unseen test set. We consider these predictions to evaluate the capacity of our model to extrapolate the knowledge learned during the training. Table

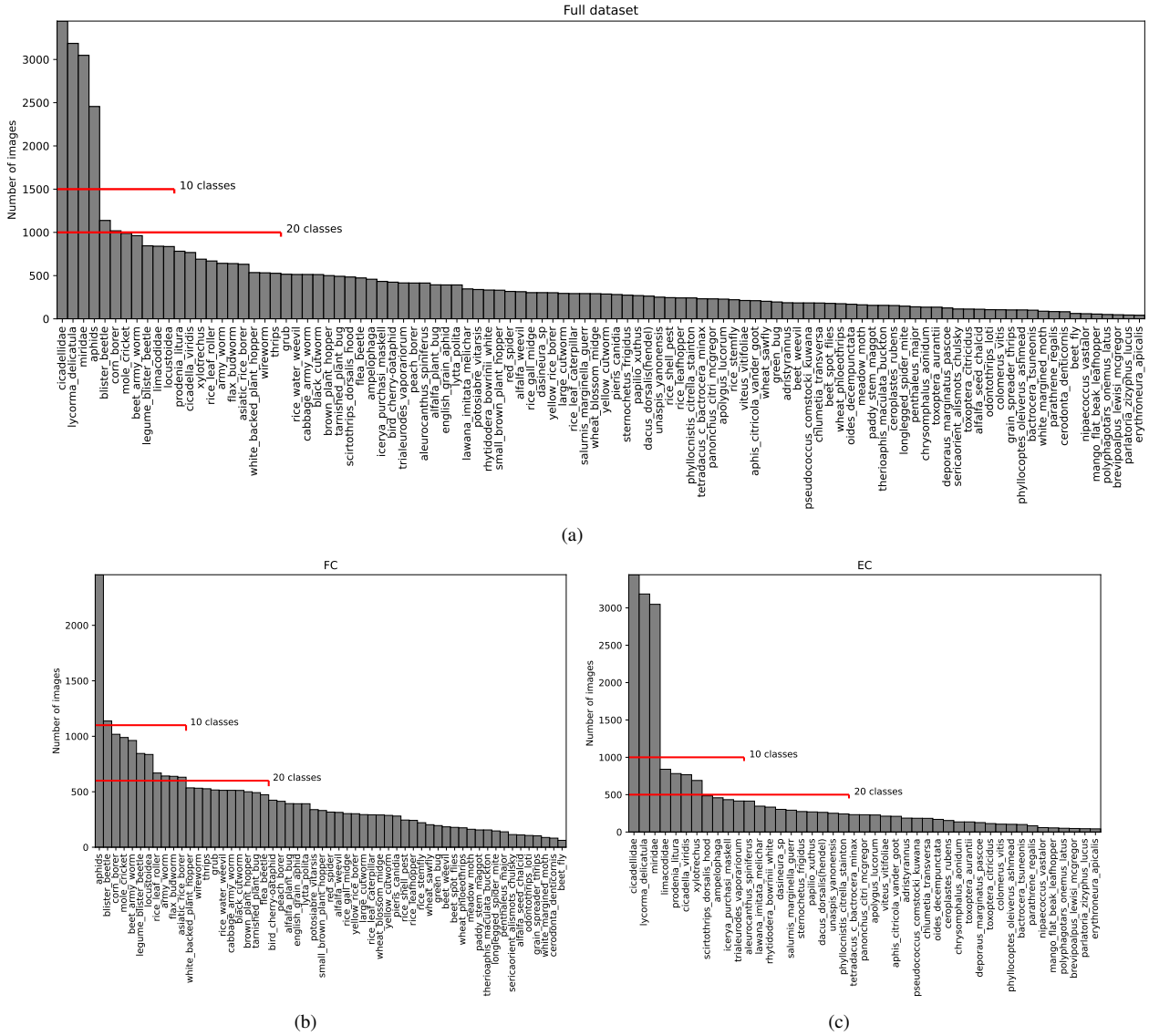

Fig. 2. Class selection for the entire dataset (a), the FC crop (b), and the EC crop type (c). Each figure shows the ten and twenty classes with more samples within.

II shows the accuracy, precision, recall, and f1-score values obtained when we submitted the test set images to our models.

In both tables, we marked in bold, for each experiment, the best result across the three training strategies. We also marked in italics the best values across the three trained architectures.

Considering the three training strategies, without data augmentation (without DA), with data augmentation but without CutMix (DA), and with data augmentation combined with CutMix (DA + CutMix), our results demonstrated that AlexNet is not suitable to be trained with data augmentation strategies. For all datasets (FC-10, FC-20, EC-10, EC-20, Full-10, and Full-20), AlexNet evaluation values tend to be worse when trained with DA and even worse with DA and CutMix combined (DA + CutMix). ResNet-50 also did not perform well when trained with data augmentation strategies. However, the differences in the evaluation metrics were smaller than in AlexNet. We may observe a slight improvement in ResNet-50 for the EC-20 and Full-20 datasets when trained with DA without CutMix.

Unlike AlexNet and ResNet-50, ViT performed well with data-augmentation strategies, mainly with data-augmentation methods combined with CutMix. For FC-10, Full-10, and EC-20, ViT performed better when trained with data augmentation and CutMix. For Full-20, training with data augmentation with and without CutMix resulted in better evaluation metrics. For EC-10, the model was trained with data augmentation without CutMix, and only for FC-20, the better model was trained without any data augmentation.

Figure 3 illustrates the test accuracy for each model considering each one of the training strategies. In the figure, one can visualize the impact of the data augmentation strategies on each model and compare them among the models. As

TABLE I
EVALUATION OVER THE VALIDATION SET FOR ALEXNET, RESNET-50, AND VIT.

VAL.			Without DA					DA					DA + CutMix				
Arch.	**Dataset**	**Classes**	**Acc.**	**Prec.**	**Rec.**	**F1**	**Epochs**	**Acc.**	**Prec.**	**Rec.**	**F1**	**Epochs**	**Acc.**	**Prec.**	**Rec.**	**F1**	**Epochs**
AlexNet	FC	**10**	**0.6867**	**0.6660**	**0.6483**	**0.6530**	69	0.6584	0.6248	0.6213	0.6143	100	0.6166	0.5859	0.5671	0.5546	104
		20	**0.6162**	**0.5826**	**0.5708**	**0.5727**	108	0.5545	0.5208	0.4976	0.4969	116	0.5103	0.4756	0.4479	0.4440	129
	EC	**10**	**0.7622**	**0.7416**	**0.7084**	**0.7231**	175	0.7197	0.6899	0.6201	0.6470	89	0.7035	0.6931	0.5764	0.6160	136
		20	**0.7055**	**0.6575**	**0.6197**	**0.6359**	97	0.6881	0.6297	0.5834	0.6004	130	0.6690	0.6183	0.5395	0.5679	181
	Full	**10**	**0.7182**	**0.7022**	**0.6796**	**0.6888**	75	0.6803	0.6557	0.6452	0.6452	77	0.6458	0.6264	0.6083	0.6081	106
		20	**0.6551**	**0.6271**	**0.5994**	**0.6095**	120	0.6039	0.5597	0.5368	0.5392	106	0.5691	0.5351	0.4936	0.4992	143
ResNet-50	FC	**10**	0.8127	0.7934	**0.7836**	**0.7877**	106	**0.8204**	**0.7988**	0.7811	0.7847	107	0.7892	0.7685	0.7489	0.7498	98
		20	**0.7288**	**0.7008**	**0.6895**	**0.6938**	124	0.6985	0.6710	0.6536	0.6570	66	0.7214	0.6964	0.6790	0.6826	173
	EC	**10**	**0.8715**	0.8464	**0.8404**	**0.8427**	60	0.8686	**0.8647**	0.8228	0.8400	98	0.8473	0.8361	0.7943	0.8109	100
		20	0.8310	0.7946	0.7566	0.7721	68	**0.8428**	**0.8022**	**0.7852**	**0.7913**	144	0.8015	0.7723	0.7054	0.7316	83
	Full	**10**	**0.8569**	**0.8461**	**0.8362**	**0.8404**	84	0.8546	0.8396	0.8299	0.8340	100	0.8489	0.8319	0.8220	0.8254	142
		20	**0.7993**	**0.7728**	**0.7558**	**0.7624**	86	0.7973	0.7652	0.7482	0.7538	148	0.7866	0.7654	0.7350	0.7450	114
ViT	FC	**10**	*0.8257*	*0.8031*	*0.7966*	*0.7987*	53	*0.8233*	*0.8002*	*0.7953*	*0.7968*	70	***0.8345***	***0.8118***	***0.8052***	***0.8071***	85
		20	*0.7423*	*0.7125*	*0.7064*	*0.7081*	58	*0.7466*	*0.7196*	*0.7133*	*0.7151*	64	***0.7641***	***0.7381***	***0.7275***	***0.7304***	118
	EC	**10**	*0.8843*	*0.8725*	*0.8451*	*0.8573*	50	***0.8954***	***0.8901***	***0.8586***	***0.8725***	61	*0.8881*	*0.8770*	*0.8557*	*0.8656*	88
		20	*0.8665*	*0.8374*	*0.8181*	*0.8265*	63	*0.8432*	*0.8067*	*0.7861*	*0.7949*	54	***0.8682***	***0.8470***	***0.8306***	***0.8373***	145
	Full	**10**	*0.8619*	*0.8452*	*0.8400*	*0.8411*	39	*0.8773*	*0.8650*	*0.8573*	*0.8607*	72	***0.8884***	***0.8779***	***0.8667***	***0.8720***	140
		20	*0.8061*	*0.7760*	*0.7622*	*0.7677*	82	*0.8136*	*0.7798*	*0.7703*	*0.7738*	95	***0.8173***	***0.7903***	***0.7780***	***0.7822***	136

TABLE II
EVALUATION OVER THE TEST SET FOR ALEXNET, RESNET-50, AND VIT.

TEST			Without DA				DA				DA + CutMix			
Arch.	**Dataset**	**Classes**	**Acc.**	**Prec.**	**Rec.**	**F1**	**Acc.**	**Prec.**	**Rec.**	**F1**	**Acc.**	**Prec.**	**Rec.**	**F1**
AlexNet	FC	**10**	**0.6948**	**0.6649**	**0.6507**	**0.6519**	0.6586	0.6257	0.6147	0.6094	0.6151	0.5795	0.5632	0.5502
		20	**0.6087**	**0.5781**	**0.5598**	**0.5647**	0.5687	0.5323	0.5161	0.5143	0.5117	0.4719	0.4462	0.4421
	EC	**10**	**0.7819**	**0.7475**	**0.7330**	**0.7396**	0.7238	0.6889	0.6343	0.6548	0.7089	0.6846	0.5999	0.6292
		20	**0.7145**	**0.6629**	**0.6269**	**0.6422**	0.6866	0.6308	0.5854	0.6013	0.6574	0.6112	0.5291	0.5582
	Full	**10**	**0.7241**	**0.7121**	**0.6959**	**0.7026**	0.6939	0.6732	0.6637	0.6641	0.6606	0.6447	0.6272	0.6277
		20	**0.6568**	**0.6196**	**0.5997**	**0.6065**	0.6117	0.5632	0.5480	0.5471	0.5783	0.5375	0.5044	0.5065
ResNet-50	FC	**10**	**0.8165**	**0.7947**	**0.7826**	**0.7874**	0.8024	0.7784	0.7637	0.7680	0.7773	0.7480	0.7323	0.7335
		20	**0.7332**	**0.7073**	**0.6976**	**0.7011**	0.7097	0.6791	0.6650	0.6681	0.7223	0.6961	0.6830	0.6860
	EC	**10**	**0.8760**	**0.8537**	**0.8493**	**0.8510**	0.8695	0.8509	0.8291	0.8383	0.8482	0.8285	0.8101	0.8165
		20	0.8393	0.8036	0.7801	0.7901	**0.8505**	**0.8066**	**0.7997**	**0.8021**	0.8102	0.7676	0.7283	0.7437
	Full	**10**	**0.8623**	**0.8516**	**0.8429**	**0.8463**	0.8598	0.8451	0.8410	0.8426	0.8572	0.8466	0.8402	0.8423
		20	0.7992	0.7680	0.7575	0.7612	**0.8033**	**0.7693**	**0.7591**	**0.7616**	0.7788	0.7514	0.7284	0.7360
ViT	FC	**10**	*0.8306*	*0.8066*	*0.7990*	*0.8022*	*0.8257*	*0.7995*	*0.7964*	*0.7970*	***0.8315***	***0.8075***	***0.8009***	***0.8031***
		20	***0.7617***	***0.7343***	***0.7295***	***0.7313***	*0.7555*	*0.7247*	*0.7248*	*0.7242*	*0.7547*	*0.7272*	*0.7206*	*0.7222*
	EC	**10**	*0.8878*	*0.8763*	*0.8601*	*0.8679*	***0.8904***	***0.8771***	***0.8611***	***0.8679***	*0.8844*	*0.8600*	*0.8590*	*0.8588*
		20	*0.8594*	*0.8253*	*0.8102*	*0.8171*	*0.8588*	*0.8304*	*0.8149*	*0.8217*	***0.8702***	***0.8406***	***0.8274***	***0.8334***
	Full	**10**	*0.8702*	*0.8562*	*0.8523*	*0.8527*	*0.8787*	*0.8692*	*0.8604*	*0.8643*	***0.8916***	***0.8813***	***0.8753***	***0.8779***
		20	*0.8164*	*0.7851*	*0.7747*	*0.7793*	***0.8242***	*0.7919*	*0.7860*	*0.7884*	***0.8242***	***0.7944***	***0.7872***	***0.7899***

discussed before, ViT is the only model that takes advantage of the data augmentation strategies, including the one we combined with data augmentation and CutMin. Besides, ViT is the best architecture for all datasets, followed closely by ResNet-50 with more distant values for the models trained with data augmentation strategies. AlexNet has the worst performance in every scenario, with values very distant from the other networks.

V. CONCLUSIONS

In this work, we performed an extensive analysis of different deep-learning models applied to classify insect pests in crop images. By considering different training strategies to train each deep learning architecture, we may find how each architecture behaves for that strategy. As we deal with a complex and highly unbalanced dataset with a large number of classes, we selected only the ten and twenty classes with more images. We also separated the full IP102 dataset into two groups in accordance with their crop classification, also considering only the ten and twenty classes with more images.

Our results lead us to conclude that transformer-based architectures performed better for this kind of problem than the CNN-based solutions. Also, the ViT models take more advantage of data augmentation strategies than the CNN counterparts. This work and the results bring important insights for building more efficient and accurate models that can be used in real applications for crop management. Systems that rapidly return an accurate classification of pests found in plantations will play an important role in agriculture production. Enabling fast responses and minimizing risks even without the presence of an expert.

Future works include evaluation of other deep learning-based architectures, looking for more adequate data augmentation strategies for each architecture. Training and evaluating strategies that work well with a large set of pests, considering even training and predictions with other datasets.

ACKNOWLEDGMENT

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

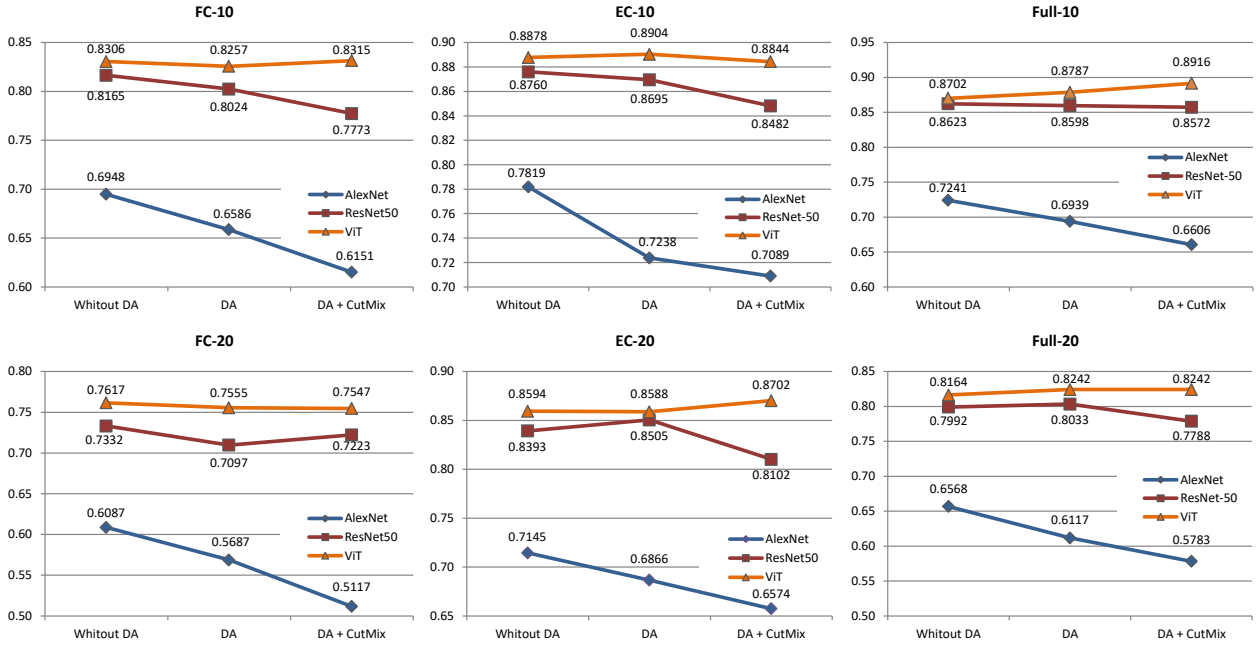

Fig. 3. Line charts presenting the accuracies of the models when trained with different data augmentation strategies. The first row is for the models trained with 10 classes, and the last row is for the models trained with 20 classes. The first column is for the FC crop classification, the second column is for the EC, and the third column is for the combination of FC and EC crop classifications.

We gratefully acknowledge the support of NVIDIA Corporation, USA with the donation of the TITAN Xp GPU used for this research.

References

[1] H. S. Pellegrina, "Trade, productivity, and the spatial organization of agriculture: Evidence from brazil," *Journal of Development Economics*, vol. 156, p. 102816, 2022.

[2] IBGE, "Pib cresce 1,9% no 1º trimestre de 2023," Available at: https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/37029-pib-cresce-1-9-no-1-trimestre-de-2023, 2023, access at: 07/28/2023.

[3] M. Preti, F. Verheggen, and S. Angeli, "Insect pest monitoring with camera-equipped traps: strengths and limitations," *Journal of Pest Science*, vol. 94, no. 2, pp. 203–217, Mar 2021.

[4] W. Albattah, M. Masood, A. Javed, M. Nawaz, and S. Albahli, "Custom cornernet: a drone-based improved deep learning technique for large-scale multiclass pest localization and classification," *Complex & Intelligent Systems*, vol. 9, no. 2, pp. 1299–1316, 2023.

[5] T. Zheng, X. Yang, J. Lv, M. Li, S. Wang, and W. Li, "An efficient mobile model for insect image classification in the field pest management," *Engineering Science and Technology, an International Journal*, vol. 39, p. 101335, 2023.

[6] Agrobot. (2020) Agrobot - agricultura robótica. [Online]. Available: https://www.agrobot.com/

[7] F. Ren, W. Liu, and G. Wu, "Feature reuse residual networks for insect pest recognition," *IEEE access*, vol. 7, pp. 122 758–122 768, 2019.

[8] H. T. Ung, H. Q. Ung, and B. T. Nguyen, "An efficient insect pest classification using multiple convolutional neural network based models," *arXiv preprint arXiv:2107.12189*, 2021.

[9] N. Ullah, J. A. Khan, L. A. Alharbi, A. Raza, W. Khan, and I. Ahmad, "An Efficient Approach for Crops Pests Recognition and Classification Based on Novel DeepPestNet Deep Learning Model," *IEEE Access*, vol. 10, pp. 73 019–73 032, 2022.

[10] S. Li, H. Wang, C. Zhang, and J. Liu, "A self-attention feature fusion model for rice pest detection," *IEEE Access*, vol. 10, pp. 84 063–84 077, 2022.

[11] L. Nanni, A. Manfè, G. Maguolo, A. Lumini, and S. Brahnam, "High performing ensemble of convolutional neural networks for insect pest image detection," *Ecological Informatics*, vol. 67, p. 101515, 2022.

[12] J. An, Y. Du, P. Hong, L. Zhang, and X. Weng, "Insect recognition based on complementary features from multiple views," *Scientific Reports*, vol. 13, no. 1, p. 2966, 2023.

[13] L. Zhang, C. Zhao, Y. Feng, and D. Li, "Pests identification of ip102 by yolov5 embedded with the novel lightweight module," *Agronomy*, vol. 13, no. 6, p. 1583, 2023.

[14] Q. Guo, C. Wang, D. Xiao, and Q. Huang, "A novel multi-label pest image classifier using the modified Swin Transformer and soft binary cross entropy loss," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 107060, 2023.

[15] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, "Ip102: A large-scale benchmark dataset for insect pest recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8787–8796.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[19] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *Pattern Recognition*, vol. 137, p. 109347, 2023.

[20] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.