www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 15(2), May-Aug, 13-25. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2024.v15i2.462

Evaluating CNN Models and Optimization Techniques for Quality Classification of Dried Chili Peppers (Capsicum annuum L.)

Carlos Guerrero-Mendez[1], David Navarro-Solís[2], Tonatiuh Saucedo-Anaya[,1], Daniela Lopez-Betancur[*,1], Luis Silva-Acosta[1], Antonio Robles-Guerrero[2], Salvador Gómez-Jiménez[2]*
[1] Universidad Autónoma de Zacatecas, Unidad Académica de Ciencia y Tecnología de la Luz y la Materia (LUMAT), Zacatecas, México.
[2] Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería, Zacatecas, México.
tsaucedo@uaz.edu.mx, danielalopez106@uaz.edu.mx

Abstract. This paper analyzes Convolutional Neural Network (CNN) models for classifying dried chili pepper quality. The models categorize images into five categories: “Extra”, “First Class”, “Second Class”, “Trash”, and “Empty”, each representing different qualities and scenarios in a sorting machine. We compared architectures from the Torchvision library, including ResNet, ResNeXt, Wide_ResNet, and RegNet using Transfer Learning (TL) in a feature extraction approach. All models employ residual blocks, an innovative technique enhancing deep learning performance. The models were evaluated using crossvalidation and metrics such as Precision, Recall, Specificity, F1-score, Geometric_mean, Index of Balanced Accuracy, and the Matthews Correlation Coefficient. They were trained using SGD, Adagrad, and Adam optimizers. Our findings suggest that ResNet-152, trained with the Adagrad optimizer, achieved the highest mean validation accuracy of 96.62%. The selected model can assist agricultural producers in classifying their products according to international standards.

Keywords: Deep learning in agricultural products, Dried chili peppers classification, Visual algorithm in sorting machines, CNN model evaluation.

Article Info
Received May 1, 2024
Accepted June 1, 2024

1 Introduction

Around the world, dried chili pepper is a common spice in a variety of foods. It is mainly used in Asian cuisine, but it is also used in Indian and Middle Eastern cuisine, as well as in Mexican food. Given its relevance in soups, sauces, dye and other applications, peppers play a pivotal role in the global agricultural economy. As a result of high demand, the pepper market is one of the fastest-growing food markets worldwide. In 2017, the global pepper production was estimated to be around 36,092,631 million tons, with China producing the highest quantity worldwide (17,821,238 tons), followed by Mexico (3,296,875 tons) (Kittler et al., 2016; Russo, 2012). The quality of dried chilies depends on several factors, such as their size (fragmented or not) and uniform color. Discoloration or brown spots are signs of poor quality. Mexico is one of the main centers of origin and dispersion of the Capsicum genus and is the center of origin of the annuum species that has generated a great diversity of types of chili peppers. Dried chili peppers are divided into three quality categories according to the Mexican Official Norm (NMX-FF-107/1-SCFI-2014): Extra, First, and Second Class (*NMX-FF-107/1-SCFI-2014*, s. f.). However, the quality of chili peppers Capsicum annuum L. (better known as "Mirasol" or “Guajillo”) is often graded and sold based on the personal experience of the buyer and seller, leading to disagreements or inequality in the negotiation of this agricultural product.

The Food industry is continually changing. One of the biggest changes in this area has been the introduction of automation in the selection of foods, whether fresh or dehydrated, which has been used to improve and guarantee the quality and efficiency of food processing operations. Automation has been used in food sorting to boost productivity, save operating costs, and improve quality in food firms. For chili pepper marketers, sorting dried chili peppers is a difficult, labor-intensive, and time-consuming task, so advancements in sorting technologies can be quite appealing.

Artificial Intelligence (AI) is among the most promising technological advances. The goal of AI, a subfield of computer science, is to build digital tools with human-like intelligence. Deep Learning (DL) is one area of AI that has shown notable progress in recent years. DL has applications in many fields, including medicine (Lopez-Betancur et al., 2021; Ortiz-Rodriguez et al., 2018; Sarvamangala & Kulkarni, 2022), agriculture (Maeda-Gutierrez et al., 2020; Too et al., 2019), food processing (Naranjo-Torres et al., 2020), physics (Guerrero-Mendez et al., 2020; Xu et al., 2021), as well as numerous others (Z. Li et al., 2022; Lopez-Betancur et al., 2022; Wang et al., 2019). Central to these advancements in DL are optimization algorithms. These algorithms play a pivotal role in training DL models. They adjust the model parameters to minimize the loss function, effectively "learning" from the data. The choice of optimizer can significantly impact the performance and precision of the model (Kingma & Ba, 2014; Reddi et al., 2019). Thus, the careful selection and tuning of these optimizers is crucial in developing efficient and precise DL models, pushing the boundaries of what AI can achieve in various fields.

A number of visual inspection systems have been created recently for the food industry and precision agriculture with the goal of cutting down on the time and expense of manual inspection. Computer vision and image analysis are typically used by sorting machines or visual inspection systems to identify product irregularities without the need for human participation. Modern sorting machines are almost as good as skilled human inspectors in terms of performance, using machine learning algorithms to automatically identify abnormal products or pin-point specific flaws with high accuracy.

Dry chili pepper sorting devices have been developed using AI and DL algorithms. Chili samples can be separated using sorting machines based on the size, color, shape, or other characteristics depending on the product. Farmers are very interested in the development of these machines because they will be able to classify their produce more efficiently.

Owing to the critical significance of pepper classification, a number of innovative studies, methodologies, and approaches have been created. Images of red chili peppers were divided into two categories in a study by Purwaningsih et al. (Purwaningsih et al., 2018): worth in chili and no worth in chili (feasible and not feasible). Using a smartphone, they took 80 pictures of the two high-quality varieties of chiles. Ten photos were set aside for validation and seventy for training from this database. Using the validation dataset, the authors' basic CNN model produced an 80% classification accuracy. On the other hand, underfitting or overfitting of the CNN model is frequently observed when the model is trained using a limited dataset.

A method for identifying the "chile" (Capsicum frustecens) and its blossom was created by Saad et al. (Saad et al., 2020). To train and validate their algorithm, they took five hundred pictures of chili plants, each with several target items. For the object detection approach, the authors used the Faster Regions with ResNet-50 (CNN model) as a feature extractor. The detection confidence level attained by Saad et al. was 65%.

Chili peppers were identified and categorized using the You Only Look Once (YOLO) version 3 object detection algorithm in research by Herdiyeni (Herdiyeni et al., 2020). Based on a set of criteria from the chili samples, the authors divided the samples into two classes, A and B. They captured the photos with a commercial smartphone. There are five peppers in each of the 100 images in the dataset. A 20% portion was used for validation and the remaining 80% for training the image dataset. To train the detector, 10,000 iterations were used. 99.4% accuracy in classification tasks and 100% accuracy in object detection were attained by its object detector algorithm. The authors also looked at classification in situations where red chili peppers overlapped, and they were 75.6% accurate.

Furthermore, Cruz-Dominguez (Cruz-Domínguez et al., 2021) used a basic artificial neural network to develop a classification technique for dried chiles. The authors developed the classification task and obtained the class of the chili as the output by computing and using the histogram of the image of the chili as the input of a multilayer perceptron. Their accuracy rate in the classification results was 82.7%. The authors of this research are motivated to assess the effectiveness of a set of cutting-edge architectures that employ residual blocks. These architectures are part of Torchvision, an open-source computer vision package from the Torch machine learning library. This paper describes an analysis of a set of CNN models that can be used to sort dried chili peppers. Model comparison and analysis are carried out using a set of statistical metrics that are frequently applied in DL. Furthermore, the CNN models were trained using several optimizer algorithms. These optimization algorithms are techniques employed to adjust the parameters of a CNN model to minimize the loss function and improve the accuracy of the model.

2 Methods, Techniques, and Instruments

This section describes the use of CNN models for the image classification task of dried chili pepper quality grades. It also details of the models and optimizers used, the methods applied, and the performance metrics employed in this study.

2.1 CNN Models

CNN models are a type of artificial neural network inspired by the structure of the human visual system and are commonly implemented in computer vision tasks (Q. Li et al., 2014). These networks are composed of multiple layers that work together to process images (See Fig. 1). Traditional CNN architectures consist of stacked convolutional layers, while newer architectures explore novel ways of constructing these layers to improve learning efficiency. The performance of CNN models can vary depending on the specific features they are trained to recognize. Therefore, it is important to compare different CNN models to determine the most optimal architecture for the specific task of classifying the quality of dried chili peppers.

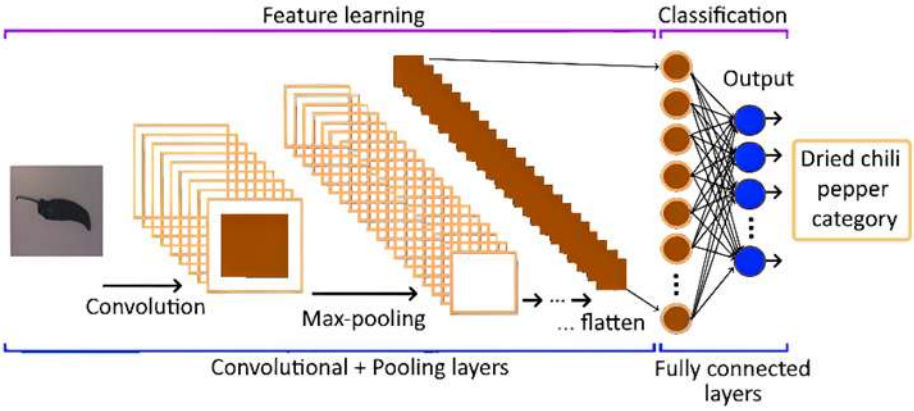

Fig. 1. Architecture of a general CNN to classify dried chili pepper images.

In order to compare the latest and best models in image classification tasks, particularly those employing residual blocks, the authors implemented some of the most accurate models reported on the Torchvision website. The subsequent part of this section introduces the ResNet, ResNeXt, Wide ResNet, and RegNet CNN models that were utilized. These models, all of which utilize residual blocks, are highly accurate, efficient, and scalable, and have been employed in a wide variety of computer vision applications (Bello et al., 2021).

ResNet (ResNet-152). ResNet, an acronym for Residual Network, is a CNN architecture that introduces the concept of skip connections (also known as shortcut connections). This innovation allows for the creation of deeper networks with numerous layers, effectively circumventing the vanishing/exploding gradient problem (He et al., 2016). ResNet emerged as the winner of the 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a competition that evaluates algorithms for large-scale image classification (Deng et al., 2009). In this research, we utilized the ResNet-152 version, signifying that the model architecture is comprised of 152 layers.

ResNext (ResNext-101). The ResNeXt model, proposed by Xie et al. (Xie et al., 2017), also implements the skip connection from the previous block to the next block (like ResNet) and aggregates a set of transformations. This new dimension, known as “cardinality”, refers to the size of the set of transformations or independent paths. The idea is to stack the same transformation blocks inside the residual block. Experiments have shown that accuracy can be improved more effectively by increasing the cardinality than by deepening or widening the model. ResNeXt was proposed by Facebook AI Research in 2017 and was designed for image classification tasks. Although ResNeXt did not win the ILSVRC 2016 challenge, it has proven to be an effective model for image classification. In this research, we used the ResNeXt-101-32x8d version, which means that the model architecture is 152 layers deep, has a cardinality of 32, and a base width of 8.

Wide ResNet (Wide resnet-101-2). Wide ResNet is an expanded and modified version of the ResNet model and was introduced by Zagoruyko and Komodakis (Zagoruyko & Komodakis, 2016) to address the problem of diminishing feature reuse and long training time caused by the increasing number of stacked layers in residual networks. The creators of Wide ResNet reduce the depth and increase the width of residual networks using wide residual blocks. Simply put, Wide ResNet has a greater number of channels compared to ResNet. For instance, the models wide_resnet50_2 and wide_resnet101_2 have a greater number of channels in the internal 3x3 convolution.

RegNet (x_32gf and y_32gf). In 2020, Facebook AI researchers Radosavovic et al. (Radosavovic et al., 2020) introduced a novel network design paradigm known as RegNet in their paper “Designing Network Design Spaces”. This approach presents a lowdimensional design space that yields simple, fast, and versatile networks. The design space combines benefits of manual design and Neural Architecture Search (NAS), thereby addressing limitations of traditional network design. The process involves parameterizing the population of networks within the design space. The primary objective for this project, according to the authors, is to advance understanding of network design and uncover design principles that generalize across settings. RegNet utilizes a single type of network block from the many different architectures available, such as the bottleneck block. RegNet has two variants: RegNetX, which employs the residual block of the classic ResNet, and RegNetY, which leverages squeeze-and-excite blocks. Although RegNet is not a CNN model per se, it represents a design space.

2.2 Optimization Algorithms

A critical factor that influences the performance of a CNN model is the optimization algorithm used in the training process. This algorithm attempts to identify optimal parameters that minimize the loss function, thereby increasing the ability of the model to classify input data and produce more accurate output data. Primarily, such an algorithm determines how to modify or adjust the weights of the neural network. So, with appropriate weight adjustments, the number of incorrectly predicted cases decreases. Additionally, certain optimizers help prevent model overfitting, among other significant aspects (Ayumi et al., 2016; Liu et al., 2021).

Stochastic Gradient Descent (SGD). SGD is one of the most popular algorithms in modern DL. Its goal is to minimize the loss function by iteratively updating the model parameters based on the gradients of the loss function. It does this by computing the gradient of the loss function with respect to the parameters for a single input-output case, and then adjusting the parameters in the opposite direction of the gradient, scaled by a learning rate. This process is repeated, often many times, until the algorithm converges to a set of parameters that achieve the minimum loss.

Adaptive Gradient Algorithm (Adagrad). Introduced in 2011 by John Duchi et al. (Duchi et al., 2011), the AdaGrad optimizer is an adaptive learning rate optimization algorithm that dynamically adjusts the learning rate for each model parameter based on the history of its past gradients. The main principle behind AdaGrad is to accumulate the sum of squares of previous gradients for each parameter and use this information to scale the learning rate for new updates. This effectively reduces the learning rate for parameters with frequently large gradients and increases the learning rate for parameters with infrequent or small gradients. By adapting the learning rate for each parameter individually, AdaGrad helps to accelerate convergence and improve performance, particularly for problems with sparse gradients.

Adaptive Moment Estimation (Adam). Introduced in 2015 by Diederik Kingma and Jimmy Ba (Kingma & Ba, 2014), Adam has become a popular choice due to its effectiveness and ease of implementation. Adam works by maintaining two exponentially decaying averages of the gradients: the first average tracks the mean of the gradients, and the second average tracks the variance of the gradients. These averages are then used to adjust the learning rate according to the characteristics of each parameter. In addition to adaptive learning rate adjustment, Adam also incorporates momentum. Momentum is a technique that helps to accelerate convergence by keeping track of the past gradients and using them to guide the current update. This helps to overcome the problem of “zigzagging” that can occur with SGD. Finally, Adam includes bias correction to account for the fact that the initial estimates of the mean and variance of the gradients are biased. This helps to ensure that the algorithm converges to the correct solution.

2.3 Transfer Learning (TL)

Training a CNN model from scratch is a highly computationally intensive task that requires a large number of labeled images. This can be particularly challenging for tasks like classifying the quality of dried chili peppers, where obtaining a large labeled dataset might be difficult. An effective alternative is to use the TL technique, which leverages the knowledge that a CNN model has gained from a previous training process on a largescale dataset, such as ImageNet. In TL, only a portion of the model is retrained to perform the new image classification task, significantly reducing the training time and computational resources required. This approach, known as “feature extraction”, involves freezing the weights of the early layers of the model and only training the final layers. Additionally, the final layer of the model is reshaped to match the number of output classes in the new task. Fig. 2 illustrates the TL process, showing a pre-trained model on ImageNet being reused to classify dried chili peppers.

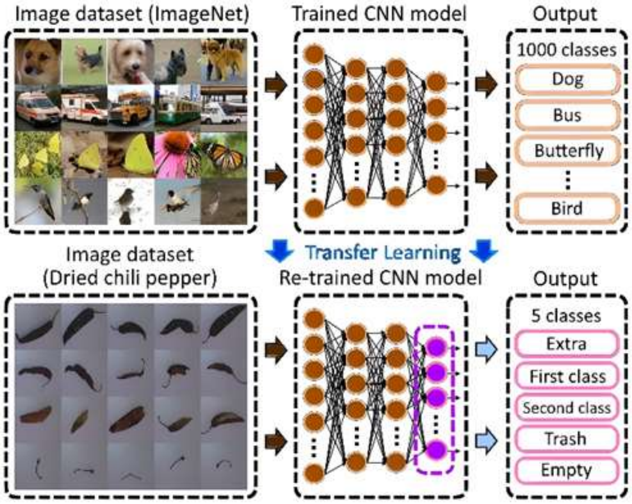

Fig. 2. Application of Transfer Learning to a CNN: Retraining the Last Layer and Modifying the Output Layer for Dried Chili Classification.

2.4 Data Acquisition

In this research, the authors adhered to the Mexican Official Norm NMX-FF-107/1-SCFI-2014 (*NMX-FF-107/1-SCFI-2014*, s. f.), which categorizes the quality of dried chili peppers into “Extra”, “First Class”, and “Second Class”, based primarily on brightness, color uniformity, size, and integrity. To facilitate the development of the sorting machine, two additional categories were introduced: “Trash”, referring to any non-chili object, and “Empty”, indicating an unoccupied conveyor belt. A diverse set of images representing each category was recorded to train the sorting machine.

The image database was generated by manually classifying several dried chili peppers with the help of an experienced product seller. Each sample was positioned 50 cm in front of a suspended camera, and both the chili samples and a consistent white background were illuminated using an RGB LED lamp situated adjacent to the camera. This setup was designed to prevent shadow casting on the samples. The image acquisition was carried out in a dark room using a Toshiba HV-F31F camera, which has a resolution of 1024 × 768 (H × V) pixels. However, the images of the dried chili peppers were captured at a resolution of 640 × 480 pixels.

The resulting dataset comprises a total of 2,866 images, distributed as follows: 564 “Extra”, 529 “First Class”, and 611 “Second Class” images of dried peppers, 541 images in the “Trash” category, and 621 images in the “Empty” category. Fig. 3 provides a representative sample from each class. To mitigate the risk of overfitting and augment the training set, data augmentation techniques were employed.

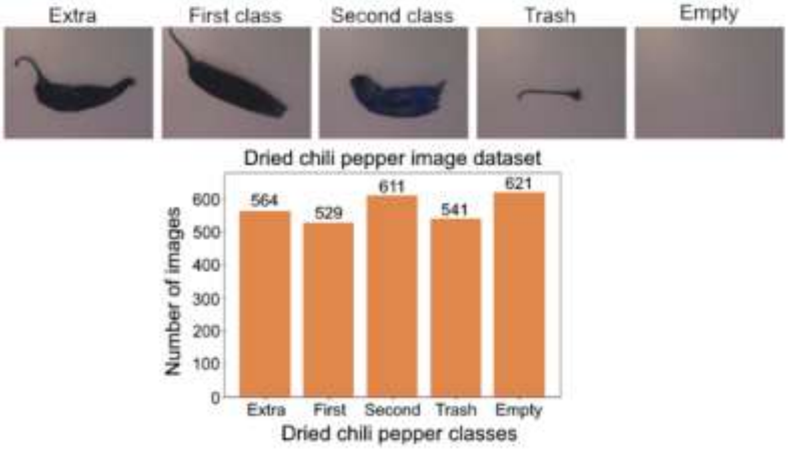

Fig. 3. Number of Image Samples for Each Dried Chili Pepper Class.

Data Augmentation. Data augmentation is a technique used to increase the diversity of a dataset by applying mathematical operations to the original images, thereby generating new image samples. This process enriches the dataset and can lead to improved model training. In this study, the authors employed three numerical transformations to augment the training dataset. These transformations were implemented using the torchvision.transforms module from the Pytorch library. The first transformation, RandomRotation, involves randomly rotating the image within a range of 0º to 90º degrees. The second transformation, RandomHorizontalFlip, entails randomly flipping the image along the horizontal axis. The final transformation, RandomVerticalFlip, involves flipping the image along the vertical axis. Fig. 4 illustrates examples of these transformations as applied to some images in the dataset.

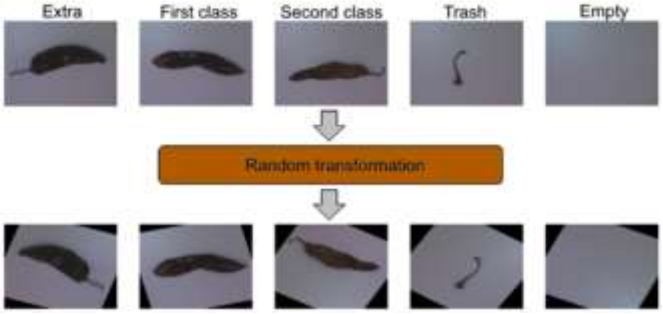

Fig. 4. Original and augmented images from each class following the data augmentation process.

2.5 Training Parameters

Key parameters in the training process are hyperparameters, which are adjustable elements in CNN models that dictate their behavior and impact on the performance of the task at hand. These “tunable” components of a model can be adjusted during the training process. Besides influencing the performance of the classification task, hyperparameters also affect the computational power and time needed to train a CNN model. The training process of a CNN model is intricate and involves a broad array of hyperparameters.

In DL, the process of identifying the best or optimal parameters for performance of a CNN model is known as optimization. This method involves an iterative process of finding values that minimize the error (loss) based on the training dataset. With the aim of implementing the most effective optimizer in a potential dried chili classification system, this study proposes a comparison and analysis of different optimization techniques in training.

One of the most critical hyperparameters is the “learning rate”. It dictates the rate at which the model adjusts the gradient of the loss function. Consequently, the learning rate controls the extent to which the model alters its predictions as it updates its results based on model error. A high learning rate causes the model to change its parameters rapidly, while a low learning rate results in slower parameter changes. The optimal approach is to select a learning rate value that allows for a correct (not overly rapid) decrease in error and finds the minimal error in the fewest number of epochs. Although this research employs various optimization algorithms that modify the learning rate, it is crucial to define the initial learning rate at which the models will commence their training.

Another significant hyperparameter is “epochs”, which refers to the number of times the entire training dataset is passed through the CNN model. However, models are trained using batches. Within a single training epoch, the “batch size” denotes the quantity of data processed by the CNN and used to update model parameters at a time until an epoch is complete. Larger batches facilitate greater computational parallelism and can often enhance performance. However, they also demand more memory and can induce latency when fed into the training function.

In DL training, the “seed” can be considered a type of hyperparameter. Although it does not directly influence learning of the model like traditional hyperparameters, the seed determines the initialization of weights and can impact the reproducibility of results and model performance. The “seed” hyperparameter serves as a starting point for a sequence of pseudorandom numbers. Given the same seed, the generator will produce the exact same sequence of numbers. This property is particularly useful for debugging and ensuring the reproducibility of model results.

The selection of hyperparameters was made without favoring any specific model or optimizer in this research. The hyperparameters utilized are detailed in Table 1.

Table 1. Hyperparameters in the training process of the models

Hyperparameter	Value
Optimization algorithms	Stochastic Gradient Descent (SGD) Adaptive Gradient Algorithm (Adagrad) Adaptive Moment Estimation (Adam)
Epochs	50
Batch size	32
Learning rate	0.001
Seed	42
Folds	5

The algorithms developed were implemented using Python 3, and the CNN models were trained utilizing PyTorch (version 2.1.0+cu118), an open-source, Python deep learning framework developed by Facebook. The training of the CNN models was performed on a Quadro P2200 GPU. We used the Torchvision package to train and evaluate the CNN models. This package provides a collection of pre-trained models and is also used to build high-quality computer vision applications. In this research, we used Torchvision version 0.16.0+cu118. Performance metrics were verified using Imbalanced-learn (Lemaître et al., 2017), an open-source Python library that provides tools for dealing with imbalanced datasets in classification tasks. An imbalanced dataset refers to a situation where the number of observations differs significantly between the classes. In other words, one class has many more samples than the other. This imbalance can lead to biased machine learning models since they tend to favor the class with more samples. Therefore, special techniques, such as those provided by Imbalanced-learn, are used to handle these imbalances. These techniques allow for more accurate evaluation of model performance, leading to more robust and fair results.

2.6 Performance Evaluation

Cross Validation. In machine learning, accuracy of a model can be high for a specific dataset but may not generalize well to other datasets due to overfitting. A more robust solution is to employ cross-validation, where the dataset is divided into multiple subsets. The model is trained and validated on these subsets, and an average accuracy value is calculated to provide a more reliable performance estimate. In this study, authors utilized 5-fold cross-validation on a single dataset to develop an optimal model for an image classification task involving dried chili peppers. This process involved partitioning the dataset into five subsets or "folds", each used once for validation while remaining folds were used for training. Fig. 5 provides a visual representation of this 5-fold cross-validation process.

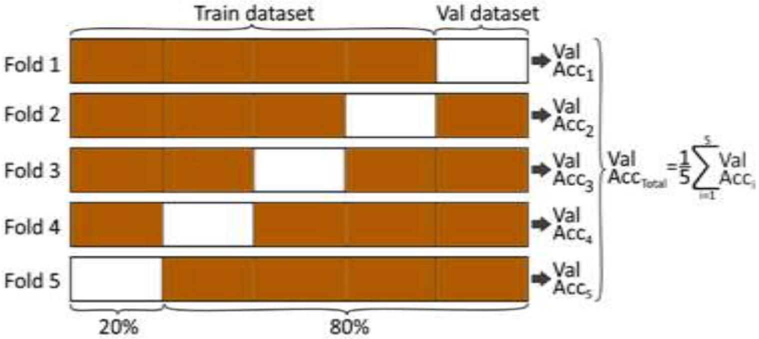

Fig. 5. K-fold Cross-Validation, with k=5.

Confusion Matrix. A confusion matrix provides a comprehensive summary of performance of a model, breaking down results into True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). This allows for a more nuanced understanding of performance of the model than accuracy alone, particularly in cases where classes are imbalanced. By analyzing the confusion matrix, one can gain insights into types of errors the model is making and potentially identify ways to improve the model. For instance, a large number of FP might indicate that the model predicts the positive class too liberally, and adjustments might be needed to make it more conservative. Conversely, a large number of FN might suggest that the model is too conservative. In the context of a CNN model, such insights can guide fine-tuning of architecture of the model or adjustment of its hyperparameters to improve its performance. Fig. 6, which shows a multiclass confusion matrix, provides a visual representation of these concepts.

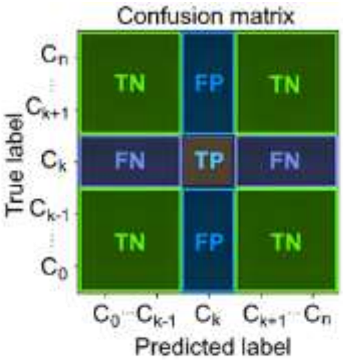

Fig. 6. Representation of a multiclass (n classes C) confusion matrix. The class of interest is k.

From four terms derived from confusion matrix, a set of important performance metrics can be calculated. These metrics include accuracy, precision, recall, specificity, F1-score, Geometric_mean (G-mean), and Index of Balanced Accuracy, also known as IBA. Furthermore, Matthews Correlation Coefficient (MCC) is a valuable metric for evaluating classifiers due to several key features. Firstly, MCC takes into account Class Balance, making it particularly informative for imbalanced classes. Secondly, Range of Values for MCC, which varies from -1 to 1, offers an intuitive interpretation of classifier performance. Lastly, ability of MCC to identify inefficiencies can help pinpoint difficulties in classifying negative class samples. These characteristics collectively contribute to importance of MCC in assessing classifier quality. Together, these metrics provide a comprehensive view of performance of the model.

The accuracy of the model is the fraction of the total samples that were correctly classified by the model. Equation (1) can be used to calculate accuracy.

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

Precision is the ability of a model to correctly classify positive elements. It is represented by Equation (2).

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

Recall (also known as sensitivity) is a metric that indicates the fraction of positive cases that the model has correctly identified as positive. The recall metric can be calculated using Equation (3).

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

Specificity is a metric that indicates the fraction of negative cases that the model has correctly predicted as negative. It is defined by Equation (4).

$$Specificity = \frac{TN}{(TN + FP)}. \tag{4}$$

The F1-score is a metric that combines precision and recall into a single score. Mathematically, it is the harmonic mean of recall and precision and is expressed as Equation (5). An F1-score of 1 indicates that the model has perfect precision and recall.

$$F1 - score = \frac{2TP}{(2TP + FP + FN)} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \, . \qquad \textbf{(5)}$$

The G-mean metric combines recall and specificity into a single metric. This metric produces a balanced value that is independent of the number of positive and negative cases. It is represented by Equation (6).

$$G - mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \, . \qquad \textbf{(6)}$$

The IBA is a metric used to evaluate the performance of a classifier on imbalanced datasets by giving more weight to the positive class (which is generally considered the most important class). In this research, a weighting factor of 0.1 is used (García et al., 2012). The IBA is represented by Equation (7).

$$IBA = \left(1 + 0.1\left(\frac{TP}{TP + FN} - \frac{TN}{TN + FP}\right)\right)\frac{TP}{TP + FN} \times \frac{TN}{TN + FP} \, . \qquad \textbf{(7)}$$

The MCC is a measure of the quality of classifications. MCC takes into account TP, TN and FP, FN, which can generally have very different sizes. The MCC is basically a correlation coefficient value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 is an average random prediction and -1 is an inverse prediction (Matthews, 1975). To calculate MCC, the Equation (8) can be used

$$MCC = \frac{(TP * TN \; - \; FP * FN)}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \, . \qquad \textbf{(8)}$$

The metrics presented in this section will be used to evaluate and compare the performance of the models in the classification task. These metrics provide a comprehensive evaluation that takes into account various aspects of model performance.

3 Results and Discussion

In the field of image classification, it is important to note that a more complex model does not necessarily yield better results. The effectiveness of the model is primarily influenced by the quality of the training database and the optimization algorithm that has been selected. As such, the careful selection of these two factors is of paramount importance for the performance of the model. In this paper, each CNN model underwent a cross-validation process using five folds. For each fold, 80% of the image dataset was used for training and the remaining 20% for validation. The highest final validation score from each fold was then used to calculate the mean validation score. A high mean validation score is indicative of the best model for the image classification task. The results revealed that the ResNet-152 model is the optimal choice for implementation in a dried chili pepper sorting machine, as it achieved a mean accuracy of 96.62%. Table 2 provides a detailed breakdown of the validation accuracy for each fold, as well as the mean fold score for each model.

Table 2. Validation Accuracy for each Fold and Mean Score for each Optimizer

Optimizer	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean Fold
			ResNet-152			
SGD	89.01%	90.40%	90.92%	90.40%	89.01%	89.95%
Adagrad	97.21%	97.03%	96.86%	96.34%	95.64%	96.62%

Adam	97.21%	90.40%	97.03%	96.68%	95.46%	95.36%
ResNeXt-101-32x8d						
SGD	89.90%	89.88%	89.01%	88.31%	88.31%	89.08%
Adagrad	95.30%	95.29%	95.81%	95.11%	94.24%	95.15%
Adam	95.64%	95.29%	94.94%	94.76%	94.59%	95.04%
Wide ResNet-101-2						
SGD	87.46%	88.13%	88.66%	88.13%	86.74%	87.82%
Adagrad	91.11%	89.88%	89.53%	90.05%	87.96%	89.71%
Adam	94.08%	95.46%	94.24%	93.89%	93.19%	94.17%
RegNet-x_32gf						
SGD	90.77%	90.92%	89.70%	90.40%	89.01%	90.16%
Adagrad	93.21%	93.37%	91.97%	91.80%	89.35%	91.94%
Adam	96.17%	96.34%	96.68%	96.68%	95.64%	96.30%
RegNet-y_32gf						
SGD	90.42%	91.10%	92.50%	90.23%	90.05%	90.86%
Adagrad	91.81%	89.88%	89.53%	88.31%	85.69%	89.04%
Adam	96.17%	95.81%	96.86%	96.86%	95.99%	96.34%

In the evaluation of the ResNet-152 model with different optimizers, Adagrad proved to be the most effective, achieving a mean fold (mean accuracy) of 96.62%. Although Adam reached the same maximum accuracy as Adagrad in Fold 1 (97.21%), it showed significant variability in the following folds, resulting in a lower mean fold of 95.36%. On the other hand, SGD maintained a constant but lower performance, with a mean fold of 89.95%. Importantly, in the fold results of the ResNet-152 model, values above 97% were achieved, a milestone not achieved with the other models and optimizers compared in this research. These results suggest that Adagrad offers more consistent performance, while Adam may be more sensitive to the specific data of each fold.

In the evaluation of the ResNeXt-101-32x8d model. Adagrad achieved the highest mean fold accuracy of 95.15%, closely followed by Adam with 95.04%, and finally SGD with 89.08%. Notably, Adam and Adagrad achieved very similar accuracies across all folds, with Adam slightly outperforming Adagrad in Folds 1 and 2, but being surpassed by Adagrad in Folds 3, 4, and 5. Despite the high accuracies achieved by Adam and Adagrad, SGD showed consistently lower performance across all folds. These results suggest that while Adagrad and Adam can achieve high accuracies, Adagrad appears to offer more consistent performance across different folds.

During the assessment of different optimizers on the Wide ResNet-101-2 model, it was found that Adam achieved the highest mean fold accuracy of 94.17%, followed by Adagrad with 89.71%, and finally SGD with 87.82%. Across all folds, Adam consistently outperformed the other optimizers, with its lowest accuracy still higher than the highest accuracy of both Adagrad and SGD.

Referring to the RegNet-x_32gf model, it was observed that Adam outperformed the other optimizers with the highest mean fold accuracy of 96.30%. Adagrad followed with a mean fold accuracy of 91.94%. Notably, for the first time, SGD surpassed 90% in mean fold accuracy, achieving 90.16%, although this was not higher than the mean fold accuracy obtained by Adagrad. Across all folds, Adam consistently achieved the highest accuracies, surpassing 96% in all but one fold. By examining the results of the RegNet-y_32gf model, it was found that Adam achieved the highest mean fold accuracy of 96.34%. However, it’s noteworthy that for the first time, SGD, with a mean fold accuracy of 90.86%, outperformed Adagrad, which had a mean fold accuracy of 89.04%. This is particularly interesting as SGD has consistently been outperformed by other optimizers in previous models. Across all folds, Adam maintained the highest accuracies, surpassing 95% in all folds. Given these results, RegNet-y_32gf could be considered as a second option for the implementation of the image classification system after ResNet-152.

Upon examining the models Wide ResNet-101-2, RegNet-x_32gf, and RegNet-y_32gf, which are designed for image recognition and composed of several regulatory units adaptable to different domain sizes and shapes, it was found that Adam consistently achieved the highest mean fold accuracy. Specifically, Adam achieved a mean fold accuracy of 94.17% for the Wide ResNet-101-2 model, and 96.30% and 96.34% for the RegNet-x_32gf and RegNet-y_32gf models, respectively. These models, based on the design space, demonstrated greater flexibility and efficiency than conventional models (Radosavovic et al., 2020).

In contrast, the models ResNet-152 and ResNeXt-101-32x8d, which are variants of the ResNet family characterized by residual blocks with an input convolutional layer and an output convolutional layer, showed superior performance with Adagrad. These models have a fixed number of residual blocks and convolutional layers, providing them with greater stability and simplicity compared to models with more blocks or layers. They also use a simple residual design, which consists of replacing each residual block with a smaller residual block with the same architecture.

These results suggest that while Adam and Adagrad can achieve high accuracies, the choice of optimizer may depend on the specific architectural characteristics of the model, such as the number of residual blocks or the adaptability of regulatory units to different domain sizes and shapes. For models with a fixed number of residual blocks, such as ResNet-152 and ResNeXt-101-32x8d, Adag-rad could be the better choice, while for models with adaptable regulatory units, such as Wide ResNet-101-2, RegNet-x_32gf, and RegNet-y_32gf, Adam appears to be more effective. These results suggest that while Adam and Adagrad can achieve high accuracies, the choice of optimizer may depend on the specific model and data of each fold.

On the other hand, if we were to develop a dried chili pepper classification system using the model and optimizer with the highest performance, we would choose the ResNet-152 model trained with the Adagrad optimizer. The classifier, which would achieve an accuracy of 97.21%, would have the metrics of precision (Pre), recall (Rec), specificity (SPE), F1-score (F1), G-mean (Geo), and the Index of Balanced Accuracy (IBA), listed in Table 3 for each class.

Table 3. Metrics obtained with the highest model ResNet-152 – Adagrad

Metric/Class	Pre	Rec	Spe	F1	Geo	IBA
“Empty”	100%	100%	100%	100%	100%	100%
“Extra	95.41%	99.05%	98.93%	97.20%	98.99%	98.00%
“First”	94.34%	91.74%	98.71%	93.02%	95.16%	89.93%
“Second”	96.33%	95.45%	99.14%	95.89%	97.28%	94.28%
“Trash”	99.12%	99.12%	99.78%	99.12%	99.45%	98.83%
Average	97.21%	97.21%	99.35%	97.20%	98.27%	96.40%

According to the results in Table 3, the model demonstrates exceptional performance across all evaluated metrics. With a precision and recall of 97.21%, the model correctly identifies 97.21% of the samples, indicating a low rate of false positives and negatives. The specificity of 99.35% reflects a very low rate of false positives. The F1-Score of 97.20%, which is the harmonic mean of precision and recall, suggests a high balance between these two metrics. In addition, the G-Mean of 98.27% indicates good model performance across all classes. Finally, the IBA of 96.40% suggests that the model is both accurate and balanced. In summary, these metrics indicate that the ReNet-152 model has solid and reliable performance.

To further illustrate the performance of the ResNet-152 model trained with Adagrad, we present the confusion matrix corresponding to the epoch of training with the highest accuracy (see Fig 7). This matrix provides a detailed visual representation of how the model correctly and incorrectly classifies samples across each class.

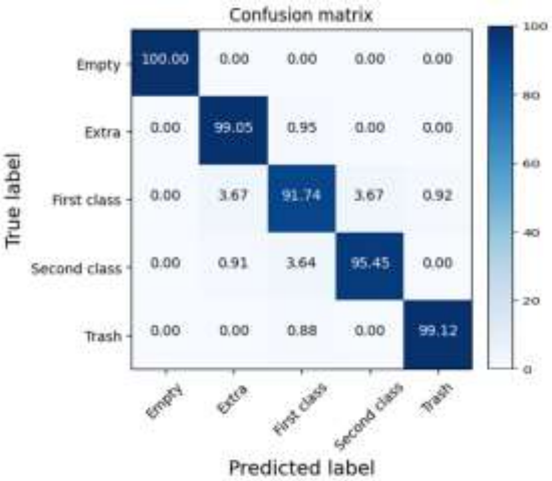

Fig. 7. Confusion Matrix of the ResNet-152 - Adagrad model.

In further discussing the results, if the ResNet-152 model trained with Adagrad is selected, a MCC value of 0.965 is obtained. This value, being remarkably close to 1, indicates an exceptional level of accuracy in the predictions made by the model. A value near 1 typically signifies that the predictions of the model align closely with the actual outcomes. This suggests that the ResNet-152 model is highly effective in classifying dried chili peppers. The high level of accuracy underscores the robustness and reliability of the model in handling this classification task.

4 Conclusions

In this research, we analyzed the performance of state-of-the-art pre-trained convolutional neural network models, all of which utilize residual blocks (ResNet, ResNeXt, Wide ResNet, and RegNet), for classifying images of quality grades of dried chilies. The aim was to identify the optimal network model for implementation in a sorting machine. Each model was trained using different optimizers in conjunction with the cross-validation method.

Our findings suggest that the choice of optimizer may depend on the specific architectural characteristics of the model. For models with a fixed number of residual blocks, such as ResNet-152 and ResNeXt-101-32x8d, Adagrad demonstrated superior performance. In contrast, for models with adaptable regulatory units, such as Wide ResNet-101-2, RegNet-x_32gf, and RegNet-y_32gf, Adam achieved the highest mean fold accuracy.

Most notably, the ResNet-152 model trained with the Adagrad optimizer achieved the highest mean fold accuracy of 96.62% among all the trained models. Furthermore, the highest accuracy obtained in a single fold using the ResNet-152 model with Adagrad was 97.21%. The performance of this model was then evaluated using advanced performance metrics, suggesting its robustness and reliability in classifying dried chili peppers. This research is expected to make a significant contribution to agriculture and food processing by providing insights into the effective use of convolutional neural network models and optimizers in image classification tasks.

References

Ayumi, V., Rere, L. M. R., Fanany, M. I., & Arymurthy, A. M. (2016). Optimization of convolutional neural network using microcanonical annealing algorithm. *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 506-511. https://doi.org/10.1109/ICACSIS.2016.7872787

Bello, I., Fedus, W., Du, X., Cubuk, E. D., Srinivas, A., Lin, T.-Y., Shlens, J., & Zoph, B. (2021). Revisiting ResNets: Improved Training and Scaling Strategies. *Advances in Neural Information Processing Systems*, *34*, 22614-22627. https://proceedings.neurips.cc/paper_files/paper/2021/hash/bef4d169d8bddd17d68303877a3ea945-Abstract.html

Cruz-Domínguez, O., Carrera-Escobedo, J. L., Guzmán-Valdivia, C. H., Ortiz-Rivera, A., García-Ruiz, M., Durán-Muñoz, H. A., Vidales-Basurto, C. A., & Castaño, V. M. (2021). A novel method for dried chili pepper classification using artificial intelligence. *Journal of Agriculture and Food Research*, *3*, 100099.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248-255.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, *12*(7). https://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf

García, V., Sánchez, J. S., & Mollineda, R. A. (2012). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems*, *25*(1), 13-21.

Guerrero-Mendez, C., Saucedo-Anaya, T., Moreno, I., Araiza-Esquivel, M., Olvera-Olvera, C., & Lopez-Betancur, D. (2020). Digital Holographic Interferometry without Phase Unwrapping by a Convolutional Neural Network for Concentration Measurements in Liquid Samples. *Applied Sciences*, *10*(14), Article 14. https://doi.org/10.3390/app10144974

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.

Herdiyeni, Y., Haristu, A., & Hardhienata, M. (2020). Chilli Quality Classification using Deep Learning. *2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA)*, 1-5.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. https://arxiv.org/abs/1412.6980

Kittler, P. G., Sucher, K. P., & Nelms, M. (2016). *Food and culture*. Cengage Learning.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, *18*(17), 1-5.

Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., & Chen, M. (2014). Medical image classification with convolutional neural network. *2014 13th international conference on control automation robotics & vision (ICARCV)*, 844-848.

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999-7019. https://doi.org/10.1109/TNNLS.2021.3084827

Liu, Z., Feng, R., Li, X., Wang, W., & Wu, X. (2021). Gradient-Sensitive Optimization for Convolutional Neural Networks. *Computational Intelligence and Neuroscience*, *2021*, e6671830. https://doi.org/10.1155/2021/6671830

Lopez-Betancur, D., Duran, R. B., Guerrero-Mendez, C., Rodriguez, R. Z., & Anaya, T. S. (2021). Comparison of Convolutional Neural Network Architectures for COVID-19 Diagnosis. *Computacion Y Sistemas*, 601-615.

Lopez-Betancur, D., Moreno, I., Guerrero-Mendez, C., Saucedo-Anaya, T., González, E., Bautista-Capetillo, C., & González-Trinidad, J. (2022). Convolutional Neural Network for Measurement of Suspended Solids and Turbidity. *Applied Sciences*, *12*(12), Article 12. https://doi.org/10.3390/app12126079

Maeda-Gutierrez, V., Galvan-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., Luna-Garcia, H., Magallanes-Quintanar, R., Guerrero Mendez, C. A., & Olvera-Olvera, C. A. (2020). Comparison of convolutional neural network architectures for classification of tomato plant diseases. *Applied Sciences*, *10*(4), 1245.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, *405*(2), 442-451. https://doi.org/10.1016/0005-2795(75)90109-9

Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R. J., Fredes, C., & Valenzuela, A. (2020). A Review of Convolutional Neural Network Applied to Fruit Image Processing. *Applied Sciences*, *10*(10), Article 10. https://doi.org/10.3390/app10103443

NMX-FF-107/1-SCFI-2014. (s. f.). https://www.dof.gob.mx/nota_detalle.php?codigo=5379404&fecha=23/01/2015#gsc.tab=0

Ortiz-Rodriguez, J. M., Guerrero-Mendez, C., Martinez-Blanco, M. del R., Castro-Tapia, S., Moreno-Lucio, M., Jaramillo-Martinez, R., & Garcia, J. A. B. (2018). Breast cancer detection by means of artificial neural networks. *Advanced Applications for Artificial Neural Networks*, 161-179.

Purwaningsih, T., Anjani, I. A., & Utami, P. B. (2018). Convolutional Neural Networks Implementation for Chili Classification. *2018 International Symposium on Advanced Intelligent Informatics (SAIN)*, 190-194.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., & Dollár, P. (2020). Designing network design spaces. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10428-10436.

Reddi, S. J., Kale, S., & Kumar, S. (2019). *On the Convergence of Adam and Beyond* (arXiv:1904.09237). arXiv. https://doi.org/10.48550/arXiv.1904.09237

Russo, V. (Ed.). (2012). *Peppers: Botany, production and uses*. CABI. https://doi.org/10.1079/9781845937676.0000

Saad, W. H. M., Karim, S. A. A., Razak, M., Radzi, S. A., & Yussof, Z. M. (2020). Classification and detection of chili and its flower using deep learning approach. *Journal of Physics: Conference Series*, *1502*(1), 012055.

Sarvamangala, D. R., & Kulkarni, R. V. (2022). Convolutional neural networks in medical image understanding: A survey. *Evolutionary Intelligence*, *15*(1), 1-22. https://doi.org/10.1007/s12065-020-00540-3

Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, *161*, 272-279.

Wang, K., Shang, C., Liu, L., Jiang, Y., Huang, D., & Yang, F. (2019). Dynamic Soft Sensor Development Based on Convolutional Neural Networks. *Industrial & Engineering Chemistry Research*, *58*(26), 11521-11531. https://doi.org/10.1021/acs.iecr.9b02513

Xie, S., Girshick, R., Dollar, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5987-5995. https://doi.org/10.1109/CVPR.2017.634

Xu, X., Tan, M., Corcoran, B., Wu, J., Boes, A., Nguyen, T. G., Chu, S. T., Little, B. E., Hicks, D. G., & Morandotti, R. (2021). 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature*, *589*(7840), 44-51.

Zagoruyko, S., & Komodakis, N. (2016). Wide Residual Networks. *Procedings of the British Machine Vision Conference 2016*, 87.1-87.12. https://doi.org/10.5244/C.30.87