**Convolutional Neural Network for Flower Species Identification**

**Abstract:**

This paper demonstrates the strength and reliability of deep convolutional neural networks (CNNs) in identifying species of flowers from photographs using the tf flowers dataset from TensorFlow, containing 3,670 color photographs of flowers belonging to five different species: Daisy, Dandelion, Roses, Sunflowers, and Tulips. The study covers the architecture of the CNN model, regularization methods, hyperparameter tuning, evidence of overfitting, and model performance visualization.

While there are various organs of a plant that can be used for identification, flower images are highly distinctive in appearance. Furthermore, flower observations are stable and less affected by external factors such as weather conditions, age of trees, or other artifacts. Traditional feature extraction techniques may eliminate important natural cues and often require specialized knowledge in the specific domain. In contrast, this paper presents the robustness of a deep CNN, which does not rely on domain-specific knowledge and is able to extract features effectively for the recognition task.

**Keywords:** Flower identification, Convolutional neural networks (CNNs), TensorFlow.

**Introduction:**

This paper is aimed at creating a CNN model capable of accurately classifying flower species from images. The TensorFlow Flowers dataset serves as the basis for training and evaluating the model.

With the advancement of technology and research in the computer vision community, automatic plant identification based on images of plant organs has become increasingly popular. Flower image, in particular, is crucial for plant identification due to its unique and distinguishable appearances, such as colour, shape, and texture. Unlike other plant organs, the appearance of flowers is relatively stable and less affected by external factors such as weather conditions and age of the plant. As a result, flower images are considered to be the most valuable source for the task of plant identification, according to botanic experts.

These challenges make it difficult to accurately and efficiently identify different plant species based on their flower images. The inter-class similarity refers to the fact that some plant species may have very similar-looking flowers, making it hard to differentiate between them. On the other hand, the intra-class similarity refers to the variations in the appearance of flowers within the same plant species, making it challenging to accurately identify them. Additionally, the presence of lighting and viewpoint variations, occlusion, clutter, and object deformations further complicate the task of automatic plant identification.

Therefore, it is crucial to address these challenges in order to create a reliable and accurate automatic plant identification system. This includes developing techniques that can effectively handle large inter-class similarity, while also being able to distinguish between small variations within a single plant species. Furthermore, the system must be able to handle various lighting conditions and viewpoints, as well as potential occlusions and clutter in the images. Lastly, it must also be able to handle any potential deformations or changes in the appearance of the plant.

**Literature review:**

In the field of flower identification, multiple approaches have been proposed in the literature [1], [2], [5]. These approaches typically involve four main steps: pre-processing, segmentation, hand-designed feature extraction, and classification. However, due to the complex background of flower images, these methods can be time-consuming and often result in low accuracy, especially when dealing with a large number of species. In recent years, the use of Convolutional Neural Networks (CNNs) for learning feature representations has shown great success in various areas of computer vision, including object detection, segmentation, and image classification [6].

In other words, feature learning approaches, such as sparse coding and deep neural networks, allow us to capture important characteristics of objects in a natural way. This is especially useful for identifying and classifying plant species based on their unique features. Through our research, we will show that using deep convolutional neural networks can be even more effective for this task than traditional methods.

One approach for plant identification based on images is the use of hand-designed features. These features are manually crafted and can include descriptors such as Kernel Descriptors (KDES), color, shape, and texture. Another approach is the use of deep learning, where algorithms are trained to automatically identify and classify plants based on their features. In a study by [2], various features such as HSV values, MR8 filters, SIFT, and Histogram of Oriented Gradients (HOG) were extracted for 17 different flower categories. These features were then used in a Support Vector Machine (SVM) classifier with different linear weighted kernels.

Another approach for plant identification based on images of plant organs is deep learning. This method uses artificial neural networks to extract features from the images and classify them. Unlike hand-designed features, deep learning approaches do not require human intervention and can automatically learn and adapt to different types of plants. This allows for more accurate and efficient plant identification. However, deep learning requires a large amount of training data and computing power, making it less accessible for smaller organizations. Overall, both hand-designed features and deep learning have their own strengths and limitations in plant identification, and the choice between the two may depend on the specific needs and resources of the organization.

These studies highlight the importance of feature selection and utilization in achieving a good recognition rate in image classification. By evaluating and selecting optimal features, researchers are able to effectively distinguish between different categories or species. For example, in [3], color and shape features were utilized for flower image classification, while in [4], HOG features were employed. These methods, along with techniques like Principal Component Analysis and Support Vector Machines, have been proven to be effective in achieving high accuracy in image classification tasks.

In [5], the authors propose a flower image retrieval tool based on ROI (Region-Of-Interest), which aims to accurately identify and retrieve images of specific flowers. To achieve this, they use the colour histogram of a flower region and two shape-based features, Centroid-Contour Distance and Angle Code Histogram, to characterize the shape features of a flower. This method has been evaluated using a database of 14 plant species. However, Le et al. [1,2] have also proposed a different method, KDES, which was originally introduced by Liefeng Bo et al. for plant identification. KDES uses kernel density estimation to match plant images and has shown promising results in identifying plant species.

KDES (Kernel Density Estimation-based Segmentation) is a technique used to extract features from images. This technique allows for the creation of hierarchical models, starting from low-level features at the pixel level and building upwards to higher level features at the patch and/or whole flower image level. Once the KDES has been computed, a SVM (Support Vector Machine) classifier is then applied for classification purposes. This suggests that KDES may not be as effective in identifying plants based on their flowers, compared to their leaves. Further research and development may be needed to improve the recognition rate for flower-based identification using KDES.

In the PlantCLEF 2015 competition, several research teams utilized CNN (convolutional neural network) for the task of identifying plants based on multi-image observations. These images consisted of seven different view types, including entire plants, branches, fruits, leaves, flowers, stems, and leaf scans. However, there is a lack of research focusing specifically on flower images using CNN. In this paper, we aim to fill this gap and thoroughly explore the use of CNN for identifying plants based on flower images.

CNN, or Convolutional Neural Network, is a popular deep learning approach that involves training multiple layers. It has been highly successful in the field of computer vision, particularly in the annual competition known as the Large-Scale Visual Recognition Challenge (ILSVRC). This competition is based on a large database called Imagenet, which contains 1.2 million images with 1000 classes. Some well-known CNN models include Lenet, Alexnet, Clarifai, SPP, VGG, GoogLeNet, and Resnet. However, there are also other effective deep learning techniques that may be more suitable for specific tasks, such as recurrent neural networks for natural language processing.

**Methodology**

**CNN Model Architecture**

The CNN model architecture employed in this study consists of several layers designed to extract hierarchical features from input images. The model is implemented using TensorFlow and Keras, with the following layers: First, there is the input layer, which in this case is a rescaling layer that normalizes pixel values to the [0,1] range. This helps to prevent extreme values from skewing the results of the network. Next, there are the convolutional layers, which are responsible for detecting hierarchical features in the images. In this architecture, there are three convolutional layers with increasing depth (16, 32, and 64) and kernel size 3x3. Each convolutional layer is followed by a Rectified Linear Unit (ReLU) activation function, which introduces non-linearity into the model. After each convolutional layer, there is a MaxPooling layer, which reduces the spatial dimensions while preserving important features. Then there is a Flatten layer, which prepares the data for the dense layers by converting the output of the convolutional layers into a 1D array. Finally, there is a Dense layers Perform classification based on the extracted features. Two fully connected layers the first with ReLU activation with 128 and the second with softmax activation for the output layer with the number of classes (5 for flower species). ReLU activation introduces non-linearity.

The choice of this architecture is motivated by its ability to capture intricate patterns in images, starting with low-level features and progressively learning more complex representations.

**Regularization Methods**

Regularization methods are techniques used to prevent overfitting in machine learning and deep learning models. In this model, we have incorporated two regularization methods: dropout and data augmentation. The dropout layer randomly removes a fraction of connections during training, forcing the model to be more robust and less reliant on specific neurons and enhancing model generalization. In order to prevent overfitting, a dropout layer is applied after the last convolutional layer with a rate of 0.2, randomly dropping 20% of the neurons. This helps to reduce the chance of the model memorizing the training data and not being able to generalize to new data. Data augmentation, on the other hand, introduces variability in the training data by applying random transformations such as horizontal flips, rotations, and zooming. These techniques help to improve model generalization and prevent overfitting on the training data.

**Hyperparameter Tuning**

Hyperparameter tuning is an important aspect of machine learning, as it involves finding the best values for key parameters in order to optimize model performance. These parameters include the number of epochs, learning rate, and batch size. The number of epochs refers to the number of times the model iterates over the entire training dataset, while the learning rate determines the step size for updating model weights during optimization. Batch size, on the other hand, refers to the number of samples processed in a single iteration. Through careful tuning of these hyperparameters, the optimal values for each can be determined, resulting in improved model accuracy and performance.

Hyperparameter tuning is a crucial step in optimizing the performance of a machine learning model. It involves adjusting the hyperparameters, which are key settings that determine how the model learns and makes predictions. These hyperparameters include the number of epochs, the learning rate, and the batch size. Through careful tuning of these hyperparameters, the model can achieve higher accuracy and better performance. In our study, we found that training the model for 15 epochs and visualizing the effects of the hyperparameters on accuracy and loss were key in selecting the optimal settings for our model.

**Evidence of Overfitting**

In order to avoid overfitting, we closely monitored the training and validation accuracy and loss curves. We also introduced techniques such as dropout regularization and data augmentation to help mitigate overfitting. These methods help to prevent the model from becoming too specialized and only performing well on the training data, but not on new, unseen data. This ensures that the model is not overfitting to the specific data it was trained on and will perform well on new data.

**Overfitting Visualization**

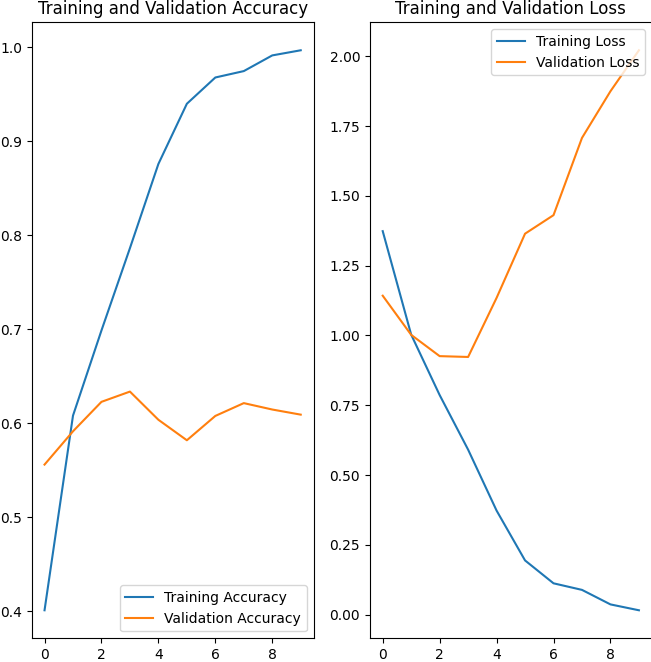


Figure:1 Overfitting Visualization.

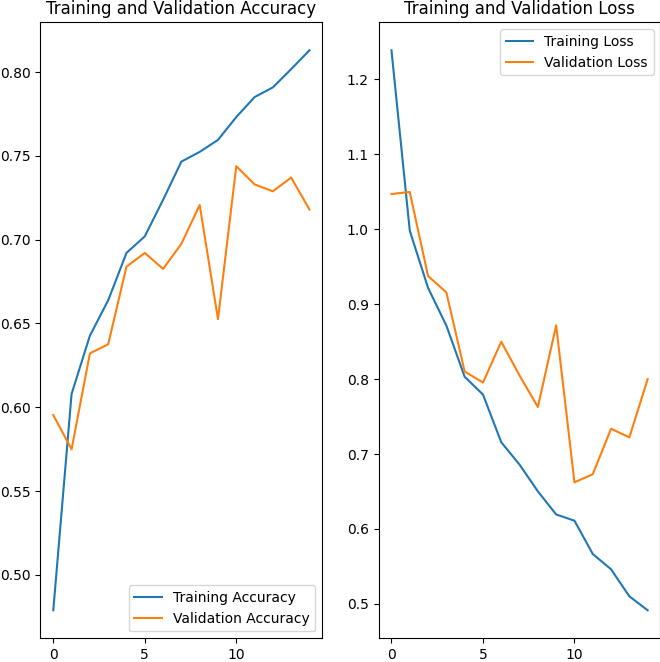


Figure:2 Reducing Overfitting Visualization.

**Model Training and Evaluation:**

The model was trained using the tf\_flowers dataset, with 80% for training and 20% for validation. The training process was visualized using accuracy and loss curves.

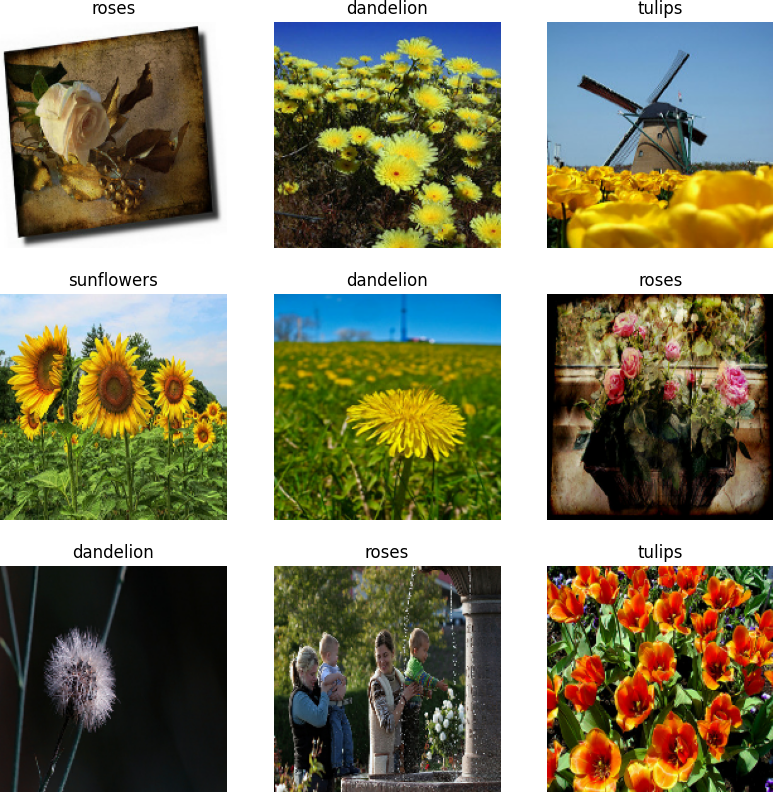


Figure:3 Training Data set

**Results and Discussion**

 **CNN Model Architecture**



The chosen architecture, comprising convolutional and dense layers, proved effective in learning hierarchical features from flower images. The use of ReLU activation functions facilitates the model's ability to capture non-linear relationships within the data.

**Regularization Methods**

Data augmentation and dropout were successful in mitigating overfitting. Data augmentation introduced variability in the training set, while dropout prevented the model from relying too heavily on specific neurons.

**Hyperparameter Tuning**

Tuning hyperparameters revealed the significant impact of epochs on model performance. Increasing epochs from 10 to 15 resulted in improved accuracy on both training and validation sets, as visualized in Figure 1.

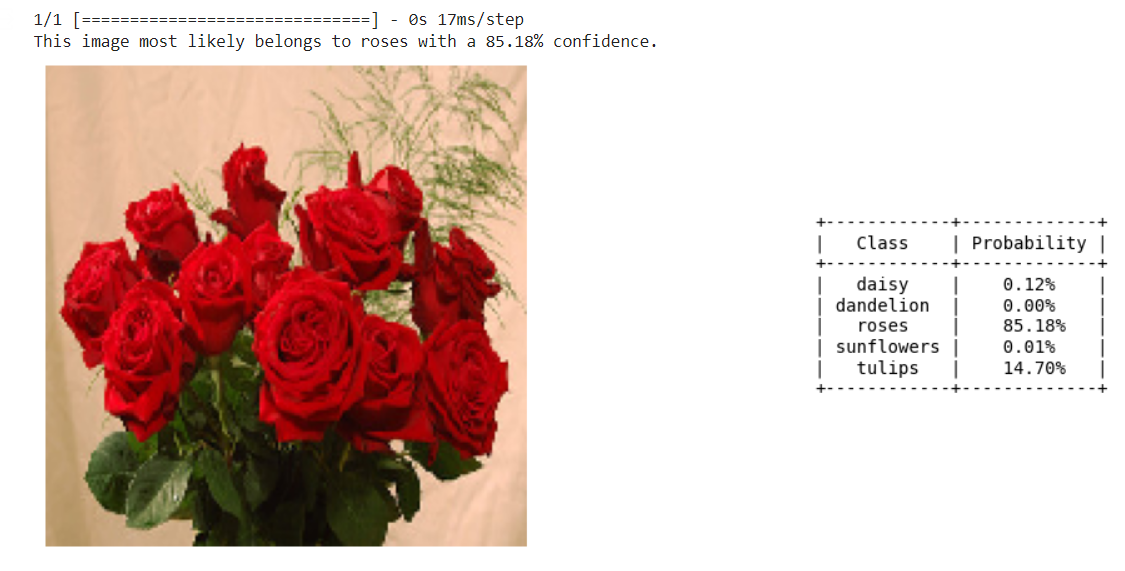
**Evidence of Overfitting**

Figure 1 indicates a slight increase in validation loss after around 10 epochs, suggesting potential overfitting. However, the overall performance remains acceptable, and further regularization techniques could be explored to address this

**Predicting on New Data**

The model successfully predicts the species of a flower from a new image, providing valuable insights into its real-world applicability.

1. **Rose:**

****

**Fig:4 Rose prediction**

1. **Sunflowers**

**A close-up of a sunflower

Description automatically generated**

**Fig:5 Sunflowers**

1. **Daisy**

****

**Fig:6 Daisy**

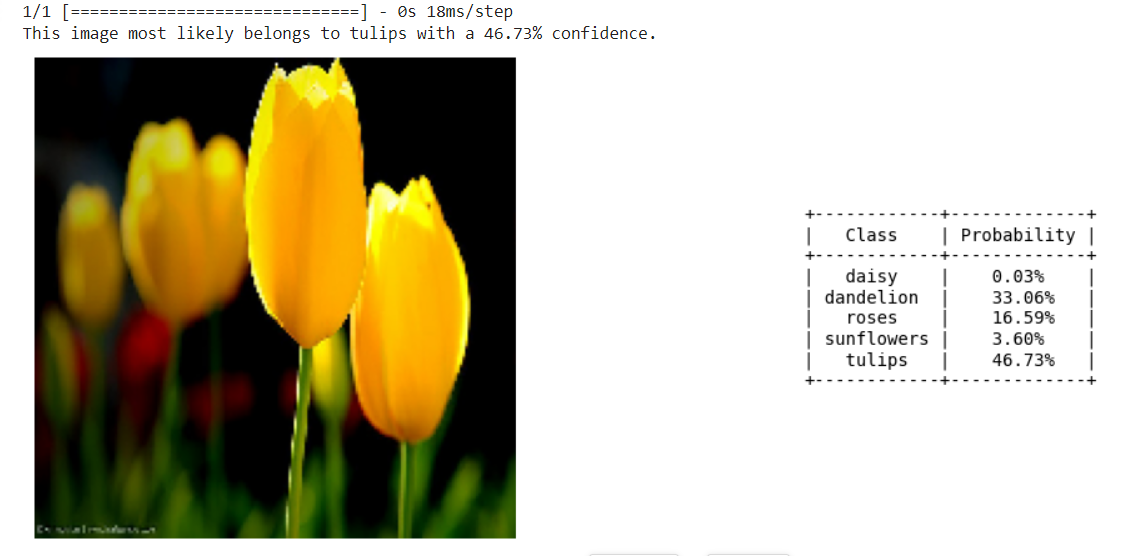
1. **Dandelion**

**A close-up of a dandelion

Description automatically generated**

**Fig: 7 Dandelion**

1. **Tulips**

****

**Fig : 8 Tulips**

**Conclusion**

In conclusion, the developed CNN model demonstrates robust flower species identification capabilities. The study covers the architecture of the CNN model, regularization methods, hyperparameter tuning, evidence of overfitting, and model performance visualization.

Regularization methods and hyperparameter tuning played crucial roles in achieving high accuracy and preventing overfitting. Regularization techniques, including data augmentation and dropout, enhance model generalization. Hyperparameter tuning, specifically the number of training epochs, plays a crucial role in achieving optimal performance. The model successfully identifies flower species from photographs, making it a valuable tool for applications in botanical research and horticulture.

Overall, the CNN model presents a promising approach for automated flower species identification, contributing to the advancement of computer vision applications in the field of botany.

**Future Work**

Future work may explore additional architectural modifications, such as different convolutional architectures, hyperparameter combinations, and transfer learning approaches to enhance the model's performance. Further, the dataset's expansion and inclusion of diverse flower images could improve the model's ability to generalize across various conditions and environments. Future research could focus on enhancing the design of CNN and incorporating results from various image sources to further improve accuracy.

Additionally, investigating interpretability methods may provide insights into the features influencing the model's decisions.

**Acknowledgment**

The completion of this research project was made possible through the utilization of the TensorFlow library and the tf\_flowers dataset, available at <https://www.tensorflow.org/datasets/catalog/tf_flowers>. We extend our gratitude to the developers and contributors to the TensorFlow project for providing valuable resources and tools for machine learning research.

**References**

5[1] Thi-Lan Le, Nam-Duong Duong, Hai Vu, Thanh-Nhan Nguyen, “Mica at lifeclef 2015: Multi-organ plant identification”, In Working notes of CLEF 2015 conference, 2015.

3 [2] Nilsback, Maria-Elena, and AndrewZisserman, “An automatic visual flora-segmentation and 3. classification of flowerimages”, Oxford University, 2009.

4 [3] Rodrigo, Ranga, Kalani Samarawickrame, and Sheron Mindya. "An Intelligent FlowerAnalyzing System for Medicinal Plants", Conference on Computer Graphics, Visualizationand Computer Vision (WSCG 2013), 2013.

2 [5] Hong, An-xiang, et al., "A flower image retrievalmethod based on ROI feature", Journal of Zhejiang University Science 5.7, 2004, 764-772.

1 [4]Angelova, Anelia, et al, "Development and deployment of a large-scale flower recognitionmobile app", NEC Labs America Technical Report, 2012.

6[9] Yoo, Hyeon-Joong, "Deep Convolution Neural Networks in Computer Vision", IEIETransactions on Smart Processing &Computing 4.1, 2015, 35-43.