FLOWER RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK (CNN)

Name: Naga Mohan Thotakura

Student Id: 23070010

GitHub Repository: repository

INTRODUTION

Deep learning is a subset of machine learning that uses multilayered neural networks, called deep neural networks, to simulate the complex decision-making power of the human brain. It consists of multiple layers of interconnected nodes, each building on the previous layer to refine and optimize the prediction or categorization.

Convolutional neural networks (CNN) are used primarily images classification applications. They can detect features and patterns within images and videos, enabling tasks such as object detection, image processing, pattern recognition and face recognition. These networks are composed of node layers containing an input layer, one or more layers and an output layer. Three main layers of CNN are convolutional layer, pooling layer and fully connected (FC) layer.

Image classification plays a crucial role in fields such as agriculture, healthcare, and environmental studies. In this report I explored CNNs for flower recognition using TensorFlow and Keras, evaluating the dataset, model architecture training process and results. Additionally compared with traditional machine learning methods like histogram of oriented Gradients (HOG) and support vector machine (SVM) which did not work that well.

DATASET

The dataset consists of the images of five flower species collected from various sources. These images vary in lighting conditions, background complexity and angles making classification more challenging. The dataset contains the following categories like daisy has 764 images, dandelion with 1052 images, rose with 784 images, sunflower with 733 images, tulip with 984images.

The total dataset comprises 4317 images which are split 80% training data 20% validation data subsets. Data augmentation techniques like rotation, zooming, flipping, shearing, rescaling which help in improving the model generation.

MODEL ARCHITECTURE

Layer (type)	Output Shape	Param #
sequential_2 (Sequential)	(None, 180, 180, 3)	0
rescaling_1 (Rescaling)	(None, 180, 180, 3)	0
conv2d_3 (Conv2D)	(None, 180, 180, 16)	448
max_pooling2d_3 (MaxPooling2D)	(None, 90, 90, 16)	0
conv2d_4 (Conv2D)	(None, 90, 90, 32)	4,640
max_pooling2d_4 (MaxPooling2D)	(None, 45, 45, 32)	0
conv2d_5 (Conv2D)	(None, 45, 45, 64)	18,496
max_pooling2d_5 (MaxPooling2D)	(None, 22, 22, 64)	0
dropout_1 (Dropout)	(None, 22, 22, 64)	0
flatten_1 (Flatten)	(None, 30976)	0
dense_2 (Dense)	(None, 128)	3,965,056
dense_3 (Dense)	(None, 5)	645

Fig 1

The model "sequential_9" which refers to a CNN built using the sequential API in Tensor flow. It has 9 layers consisting of convolutional pooling dropout and dense layers. The sequential architecture consisting of multiple layers like:

1.Con2D(32filters,3x3kernel,ReLU activation): Detects low-level features like edges.

2.MaxPooling2D(2x2): Reduces spatial dimensions, preventing overfitting.

3.Conv2D (64 filters, 3x3 kernel, ReLU activation)- identifies more complex patterns,

4.MaxPooling2D(2x2)

5.Conv2D (128 filters,3x3 kernel, ReLU activation)-Captures high-level image features.

6.Maxpooling2D(2x2)

7.Flatten-Converts feature maps into a one-dimensional array.

8.Dropout(0.2) Prevents overfitting by randomly disabling neurons during training.

9.Flatten-Converts feature maps into a one-dimensional array.

10.Dense (128 neurons, activation=ReLU)-Fully connected layer for feature interpretation.

11.Dense (5 neurons, SoftMax activation)-Produces scores for each flower class.

The model is compiled using categorical cross-entropy loss and Adam optimizer and training is conducted over 25 epochs with a batch size 32.

TRAINING AND EVALUATION

The Model is trained on the augmented dataset and validated on the test set. The key hyperparameters used in training include:

Optimizer: Adam

Loss function: Categorical cross-entropy

Batch Size:32

Epochs:20

Total Parameters:3989285(15.22 MB)

The model is trained with 20 epochs and batch size as 32 is trained on the training data and is stored with '.h5' extension in the local. Then the model is tested with sample images.

Data augmentation is done to improve model's ability to generalize and reduce overfitting. Augmentation techniques like random flipping, random rotation and random zooming are used which helps it to get exposed to variety of data.

Fig. 2

The above grouped images are used for training the model with different angles and changing resolutions.

The Adam Optimizer is an advanced optimization algorithm that dynamically adjusts the learning rate for each parameter using first and second moment estimates. It is well suited for deep learning tasks by improving convergence speed and stability.

Sparse Categorical Cross entropy is a loss function used for multi-class classification when labels are provided as integers rather than one-hot encoder vectors. It computes the probability of the correct class while it is handling the large categorical datasets.

Results

Training Accuracy:89.5%

Validation accuracy:75%

Validation loss:75.48

These results suggest that the model effectively learns features for the flower recognition with minimal overfitting.

CHALLENGES AND LIMITATIONS

Though a good accuracy is achieved, our CNN model faces several challenges like:

1.Class Imbalance: Some classes contain more images, potentially biasing predictions.

2.small dataset: More training data could improve generalization.

3.visual similarities: Flowers with same features may be misclassified.

4.Computational Requirements: CNNs demand a goof computing power for training.

To overcome these limitations, future improvements include:

1.Transfer Learning: Using pre trained models like VGG16 or ResNet for better feature extraction.

2. Additional data augmentation: Introducing new transformations to enhance diversity.

3.Larger dataset collection: Expanding the dataset to improve classification accuracy.

Conclusion

The CNN model architecture explained in this project, effectively classifies flowers by extracting hierarchal features progressively increasing the complexity at each layer. The model leverages data augmentation, dropout layer and normalization to enhance generalization and reduce overfitting. It performed well comparing to traditional methods like SVM etc. Further research can be done on optimizing architectures and using more amount to data and using transfer learning to enhance the model performance.

REFERENCES

1.https://www.researchgate.net/publication/344249077_Flower_Identification_and_Classification_using_Computer_Vision_and_Machine_Learning_Techniques

2.https://www.researchgate.net/publication/333242381_Convolutional_Neural_Network_Layers_and_Architectures

3.https://www.researchgate.net/publication/382137893_Flowers_Images_Classification_with_Deep_Learning_A_Review

4.https://www.researchgate.net/publication/368258019_Sunflower_Image_Classification_Using_Multiclass_Support_Vector_Machine_Based_on_Histogram_Characteristics