Tiny ImageNet Classification

PRESENTED BY:

BHUVAN KARTHIK CHANNAGIRI

GAYATHRI SURESH

NIKITA VINOD MANDAL

PRESENTED ON: 18TH APRIL 2024

INTRODUCTION

Motivation:

Enhance image classification using CNNs on Tiny ImageNet dataset.

Challenge:

Low resolution (64x64 pixels) of Tiny ImageNet images.

Objectives:

-Explore efficacy of 6 Convolutional Neural Networks architectures for classification.

-Refine methodologies for data preprocessing and augmentation.

-Deepen understanding of CNN architectures and image classification techniques.

DATASET

- **ImageNet Dataset:**
- *Number of Classes:* 1,000.
- *Number of Images:*
    - *Training:* 1.2 Million, spanning the 1,000 classes.
    - *Validation:* 50,000, across the same classes.
    - *Test:* 100,000, from the training classes.

n04254777 (806)

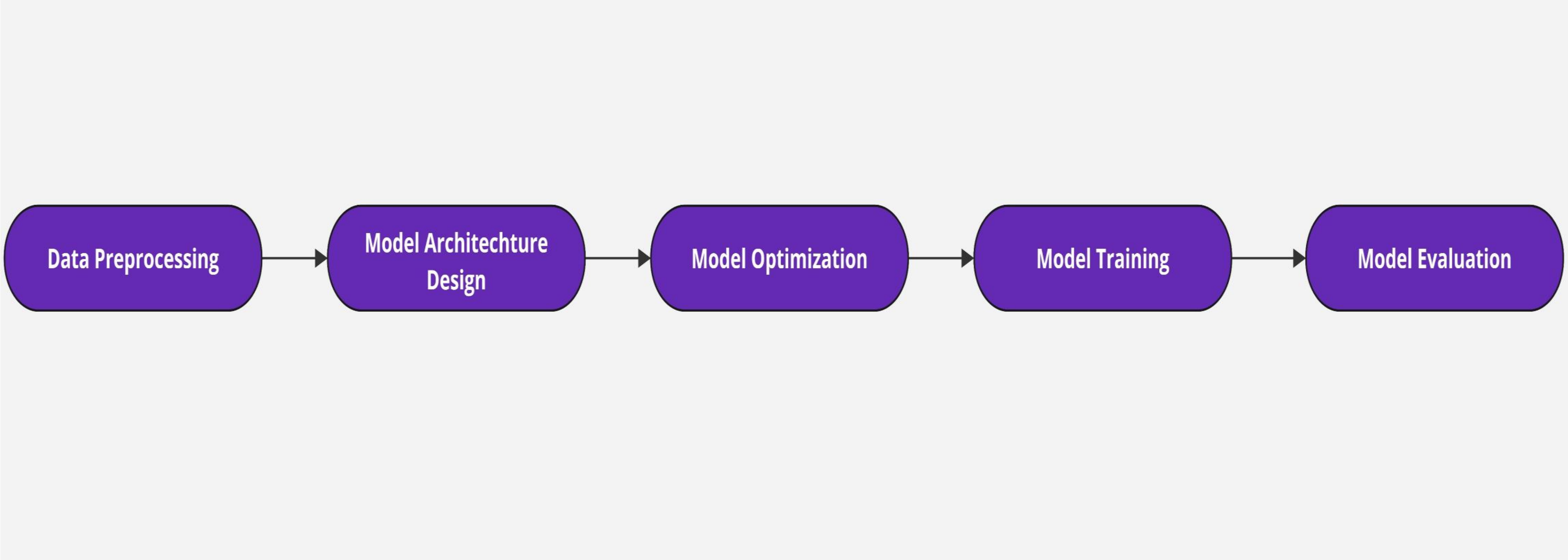
n02859443 (449)

n02107683 (239)

n01443537 (1)

- **Tiny ImageNet Dataset:**
- *Number of Classes:* 200.
- *Number of Images:*
    - *Training:* 10,00,000, distributed among the 200 classes.
    - *Validation:* 10,000, from the training classes.
    - *Test:* 10,000, for model evaluation.

METHODOLOGY



DATA PREPROCESSING AND AUGMENTATION

Rescaling:

Pixel values are scaled to 0-1 range for better convergence.

Augmentation:

-Shear, zoom, flip, rotation, and shift ranges introduce diverse variations.

-Brightness adjustment enhances model robustness under different lighting.

Fill mode:

Specifies how transformation-created areas are filled, with 'nearest' selecting the nearest pixel value.

TRAINING

- **Models Created:**
  - Five distinct Convolutional Neural Networks (CNNs) were trained
- **Activation Functions:**
  - ReLU activations are utilized within hidden layers
  - Softmax activation is employed in the output layers
- **Optimizer Selection:**
  - Adam optimizer chosen for training
  - Learning rate chosen as 0.0001
  - Alternatives assessed included RMSProp, SGD, and Nadam
- **Objective:**
  - Optimal performance sought through careful selection of optimizer and learning rate

MODELS USED

Model 1:

- Custom CNN architecture for image classification, employing convolutional and max-pooling layers.
- Includes L2 regularization, Adam optimizer and categorical cross-entropy

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 64)	18,496
max_pooling2d_1	(None, 14, 14, 64)	0
flatten	(None, 12544)	0
dense	(None, 1024)	12,846,080
dense_1 (Dense)	(None, 200)	205,000

Model 2:

- 3-layer CNN architecture with max-pooling for downsampling.
- Dropout regularization, Adam optimizer, early stopping, and learning rate reduction for optimal training.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 64)	18,496
max_pooling2d_1	(None, 14, 14, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 128)	73,856
max_pooling2d_2	(None, 6, 6, 128)	0
flatten	(None, 4608)	0
dense	(None, 512)	2,359,808
dropout	(None, 512)	0
dense_1 (Dense)	(None, 200)	102,600

MODELS USED

Model 3:

Model is structured as follows:

- Block 1, 2, 3, 4 and 5:
  - 2 convolutional layers, each with 64, 128, 256 and 512 filters respectively.
  - Batch normalization for each layer.
  - Max pooling layer with a stride of 2x2.
  - Upon flattening, three dense layers were added(4096, 1024, 200)

Layer (type)	Output Shape	Param #
Conv2D	(None, 64, 64, 64)	1,792
BatchNormalization	(None, 64, 64, 64)	256
Activation	(None, 64, 64, 64)	0
Conv2D	(None, 64, 64, 64)	36,928
BatchNormalization	(None, 64, 64, 64)	256
Activation	(None, 64, 64, 64)	0
MaxPooling2D	(None, 32, 32, 64)	0
Conv2D	(None, 32, 32, 128)	73,856
BatchNormalization	(None, 32, 32, 128)	512
Activation	(None, 32, 32, 128)	0
Conv2D	(None, 32, 32, 128)	147,584
BatchNormalization	(None, 32, 32, 128)	512
Activation	(None, 32, 32, 128)	0
MaxPooling2D	(None, 16, 16, 128)	0
Conv2D	(None, 16, 16, 256)	295,168
BatchNormalization	(None, 16, 16, 256)	1,024
Activation	(None, 16, 16, 256)	0
Conv2D	(None, 16, 16, 512)	1,180,160
BatchNormalization	(None, 16, 16, 512)	2,048
Activation	(None, 16, 16, 512)	0
MaxPooling2D	(None, 8, 8, 512)	0
Flatten	(None, 32768)	0
Dense	(None, 4096)	134,221,824
BatchNormalization	(None, 4096)	16,384
Activation	(None, 4096)	0
Dense	(None, 1024)	4,195,328
BatchNormalization	(None, 1024)	4,096
Activation	(None, 1024)	0
Dense	(None, 200)	205,000

MODELS USED

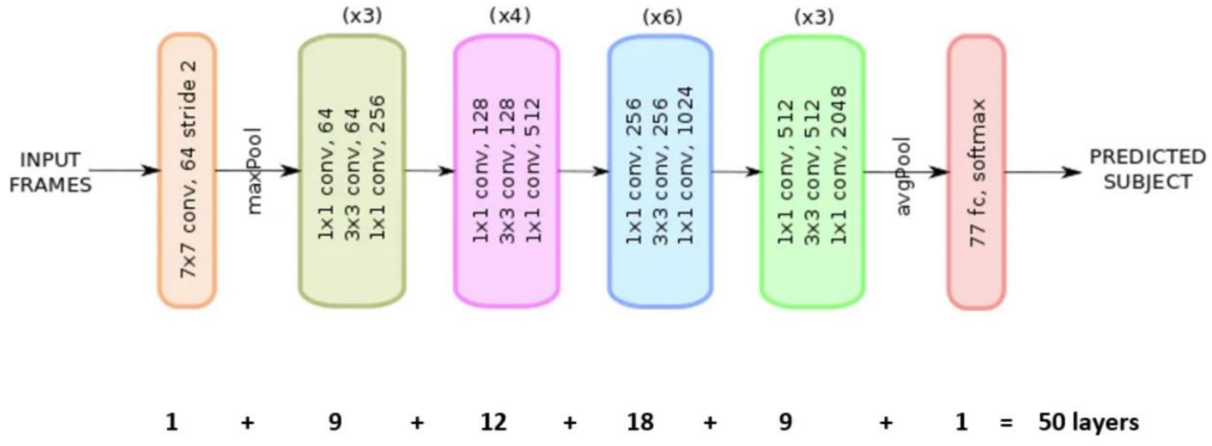
Model 4:

Model is structured as follows:

- Blocks 1, 2 and 3:
  - Convolutional layers (64 ,128 and 256 filters), batch normalization, ReLU activation, max pooling (2x2).
- Dense Layer:
  - 512 units, batch normalization, ReLU activation, dropout (rate: 0.5).
- Output Layer:
  - Dense layer with softmax activation.

Layer (type)	Output Shape	Param #
Conv2D	(None, 64, 64, 64)	1792
BatchNormalization	(None, 64, 64, 64)	256
Activation	(None, 64, 64, 64)	0
MaxPooling2D	(None, 32, 32, 64)	0
Conv2D	(None, 32, 32, 128)	73856
BatchNormalization	(None, 32, 32, 128)	512
Activation	(None, 32, 32, 128)	0
MaxPooling2D	(None, 16, 16, 128)	0
Conv2D	(None, 16, 16, 256)	295168
BatchNormalization	(None, 16, 16, 256)	1024
Activation	(None, 16, 16, 256)	0
MaxPooling2D	(None, 8, 8, 256)	0
Flatten	(None, 16384)	0
Dense	(None, 512)	8389120
BatchNormalization	(None, 512)	2048
Activation	(None, 512)	0
Dropout	(None, 512)	0
Dense	(None, 200)	102600

MODELS USED

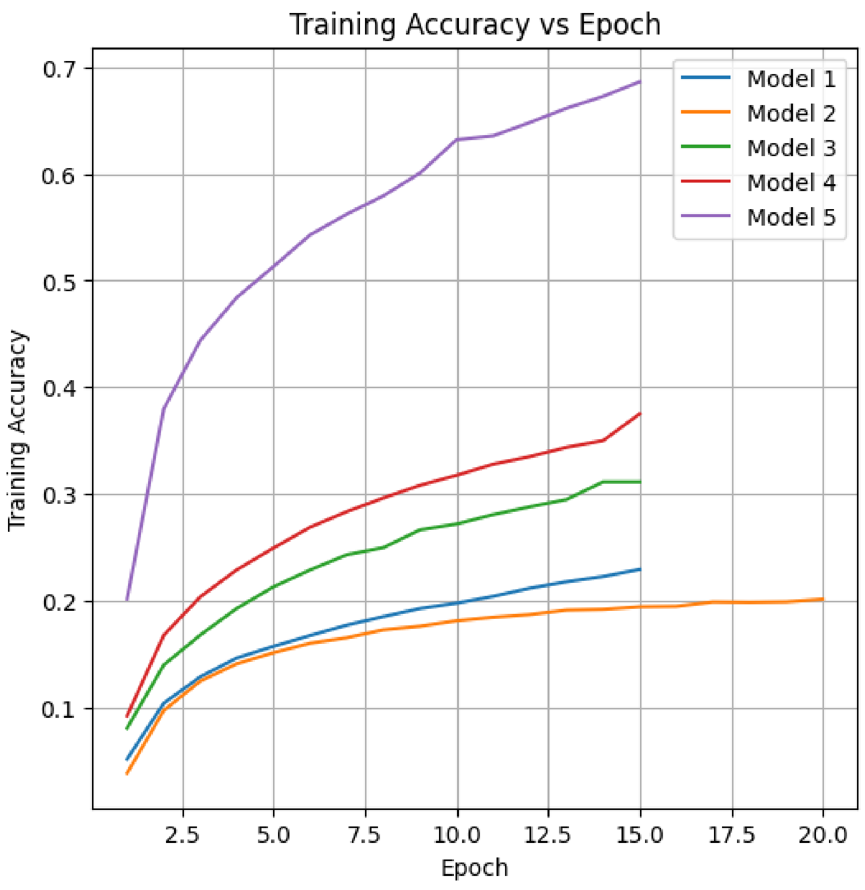


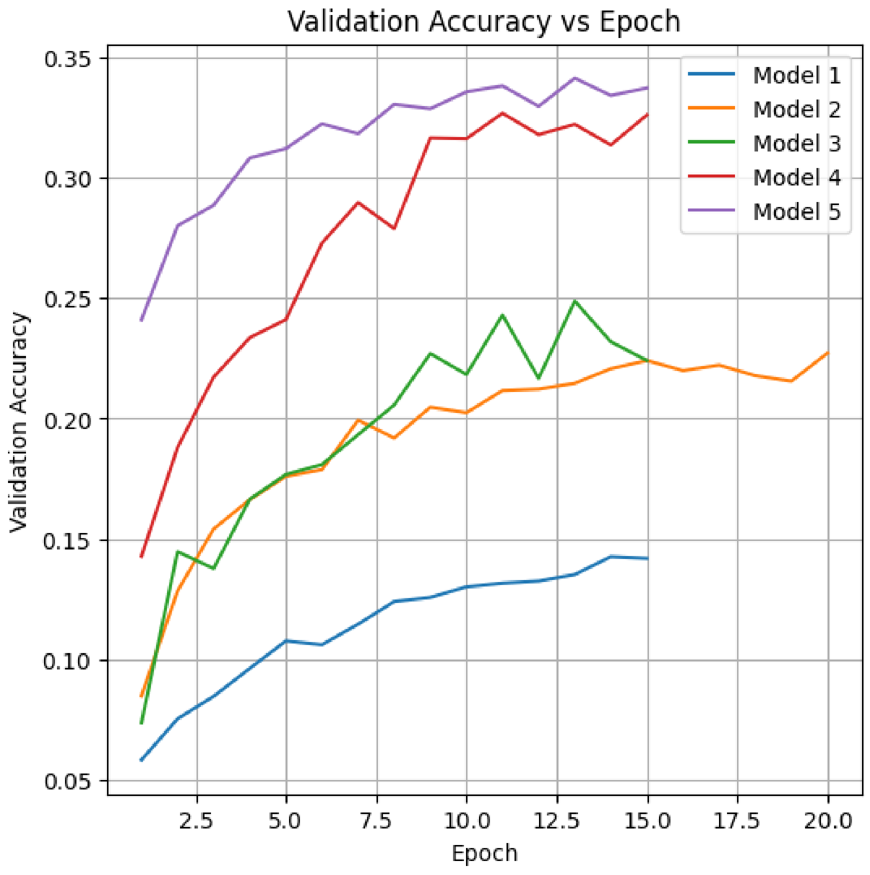
Model 5:

Model is structured as follows:

- ResNet50 Backbone:
  - Pre-trained on ImageNet data.
- Feature Extraction:
  - Global Average Pooling layer followed by a dense layer with 1024 units and ReLU activation, with dropout (rate: 0.5).
- Output Layer:
  - Dense layer with softmax activation, predicting NUM_CLASSES classes.
- Compilation Details:
  - Optimizer: Adam with a learning rate of 0.0001.
  - Loss Function: Categorical Crossentropy.

RESULTS


Training Accuracy vs Epoch
Training Accuracy
Epoch
Model 1
Model 2
Model 3
Model 4
Model 5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Validation Accuracy vs Epoch
Validation Accuracy
Epoch
Model 1
Model 2
Model 3
Model 4
Model 5
0.05
0.10
0.15
0.20
0.25
0.30
0.35
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

RESULTS

Model	Train Accuracy	Validation Accuracy
Model 1	22.96%	14.19%
Model 2	20.17%	22.72%
Model 3	29.50%	24.88%
Model 4	37.52%	38.61%
Model 5	68.61%	43.14%

CONCLUSION AND FUTURE WORK

- Deep CNNs proved effective for image classification on Tiny ImageNet, achieving up to 43.14% validation accuracy.
- Insights from various architectures highlight the importance of deeper models and regularization techniques.

- Explore advanced regularization and fine-tuning for improved accuracy.
- Investigate more intricate models like VGG16 to extract richer information from low-resolution images.

QUESTIONS ?