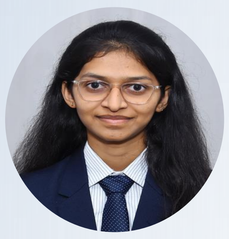
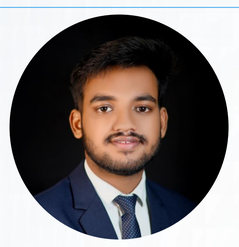
Semantic Image Clustering

EdTech Internship Program: Analytics, Data Science, & Emerging Technologies.

23 - Tech Troopers


Tanmayee Patil


Sayali Yewale


Varun Sangewar


Geeta Chate

Coordinators: Dr. Ashwin T S & Dr. Kapil B. Kadam

Mentor: Prof.Milind Vadgavi

Introduction:

- Semantic image clustering groups images based on content and meaning, emphasizing semantic relationships and context.
- This project automates clustering by combining deep learning with traditional algorithms, eliminating manual labeling.

Methodology:

- GitHub Basics.
- Keras Data and Methodology.
- EdTech Data and Methodology.

Objectives:

- To design and implement a CNN-based encoder using ResNet50 for robust feature extraction.
- To use traditional clustering algorithms, like K-Means, to group similar images semantically.

Dataset:

- CIFAR-10:
  1. 60,000 32x32 color images in 10 different classes, with 6,000 images per class.
  2. The dataset is split into 50,000 training images and 10,000 test images.
- CIFAR100:
  1. This dataset is just like the CIFAR-10, except it has 100 classes containing 600 images each.
  2. There are 500 training images and 100 testing images per class.
- IEMOCAP:
  1. It contains video-visual data from 10 actors (5 male and 5 female), was used.
  2. We extracted individual frames from the videos at a rate of 1 frame 30 second, resulting in a substantial number of images to annotate.

Models Used:

- Utilized ResNet50 and Scan algorithm in Keras for feature extraction.
- Applied K-Means clustering for the IEMOCAP dataset to group images based on semantic content.

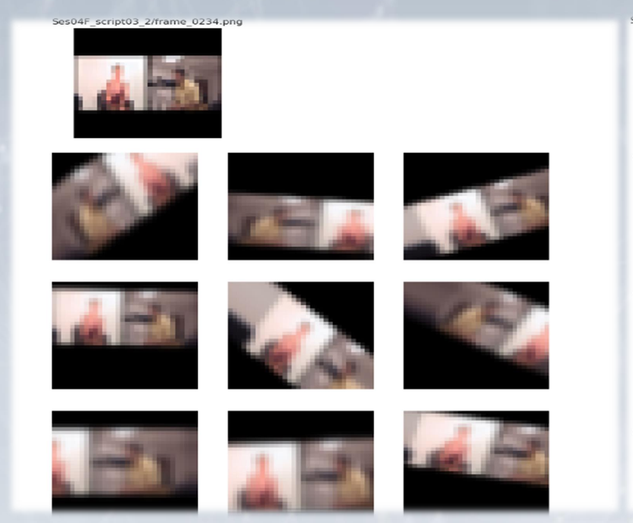


Fig. Data Augmentation

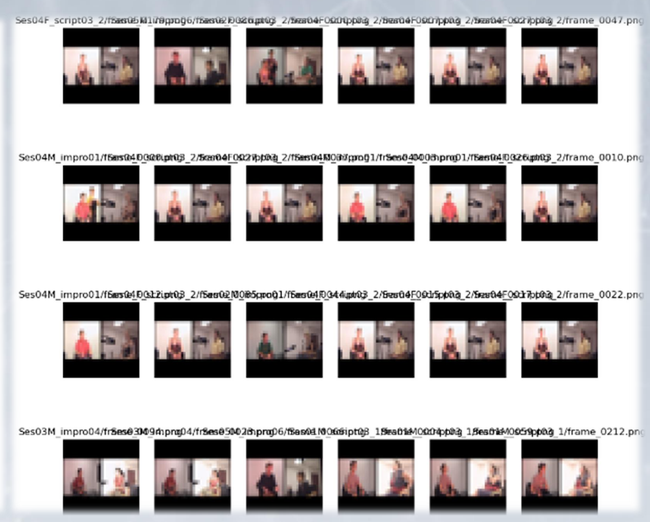
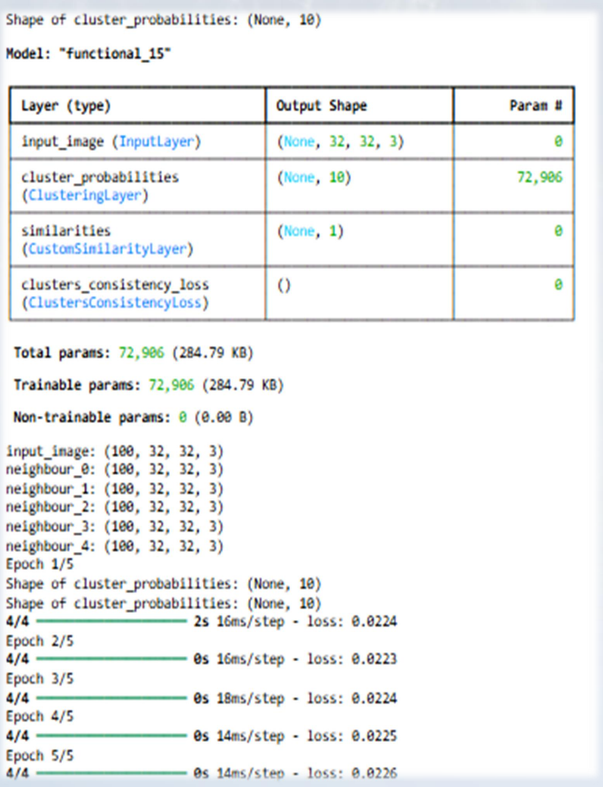


Fig. Image Clustering



```
Shape of cluster_probabilities: (None, 10)

Model: "functional_15"
```

Layer (type)	Output Shape	Param #
input_image (InputLayer)	(None, 32, 32, 3)	0
cluster_probabilities (ClusteringLayer)	(None, 10)	72,906
similarities (CustomSimilarityLayer)	(None, 1)	0
clusters_consistency_loss (ClustersConsistencyLoss)	()	0

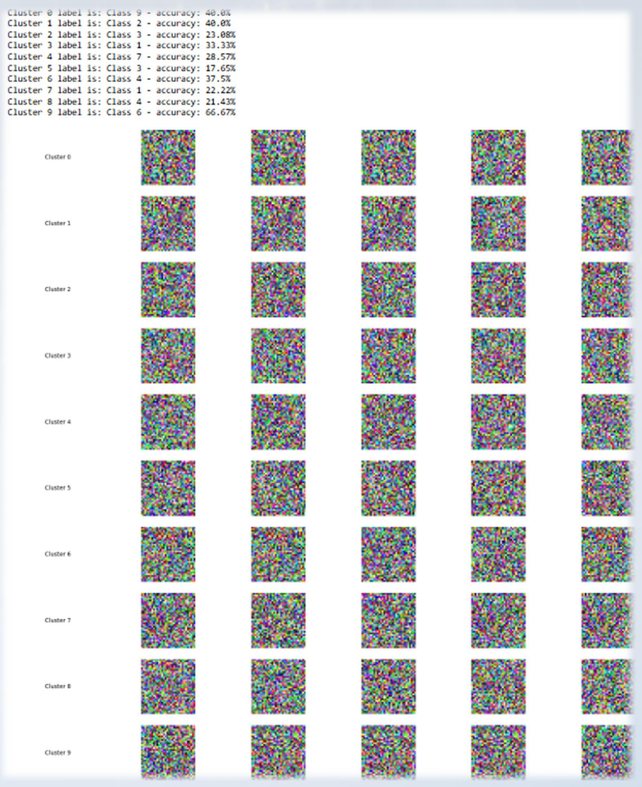
```
Total params: 72,906 (284.79 KB)

Trainable params: 72,906 (284.79 KB)

Non-trainable params: 0 (0.00 B)

input_image: (100, 32, 32, 3)
neighbour_0: (100, 32, 32, 3)
neighbour_1: (100, 32, 32, 3)
neighbour_2: (100, 32, 32, 3)
neighbour_3: (100, 32, 32, 3)
neighbour_4: (100, 32, 32, 3)
Epoch 1/5
Shape of cluster_probabilities: (None, 10)
Shape of cluster_probabilities: (None, 10)
4/4 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step - loss: 0.0224
Epoch 2/5
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 0.0223
Epoch 3/5
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.0224
Epoch 4/5
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 0.0225
Epoch 5/5
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 0.0226
```

Fig. Model Architecture and Training for Image Clustering Learning



```
Cluster 0 label is: Class 9 - accuracy: 40.0%
Cluster 1 label is: Class 2 - accuracy: 40.0%
Cluster 2 label is: Class 3 - accuracy: 23.08%
Cluster 3 label is: Class 1 - accuracy: 33.33%
Cluster 4 label is: Class 7 - accuracy: 28.57%
Cluster 5 label is: Class 3 - accuracy: 17.65%
Cluster 6 label is: Class 4 - accuracy: 37.5%
Cluster 7 label is: Class 1 - accuracy: 22.22%
Cluster 8 label is: Class 4 - accuracy: 21.43%
Cluster 9 label is: Class 6 - accuracy: 66.67%
```

Fig. Clustering Results and Accuracy Analysis

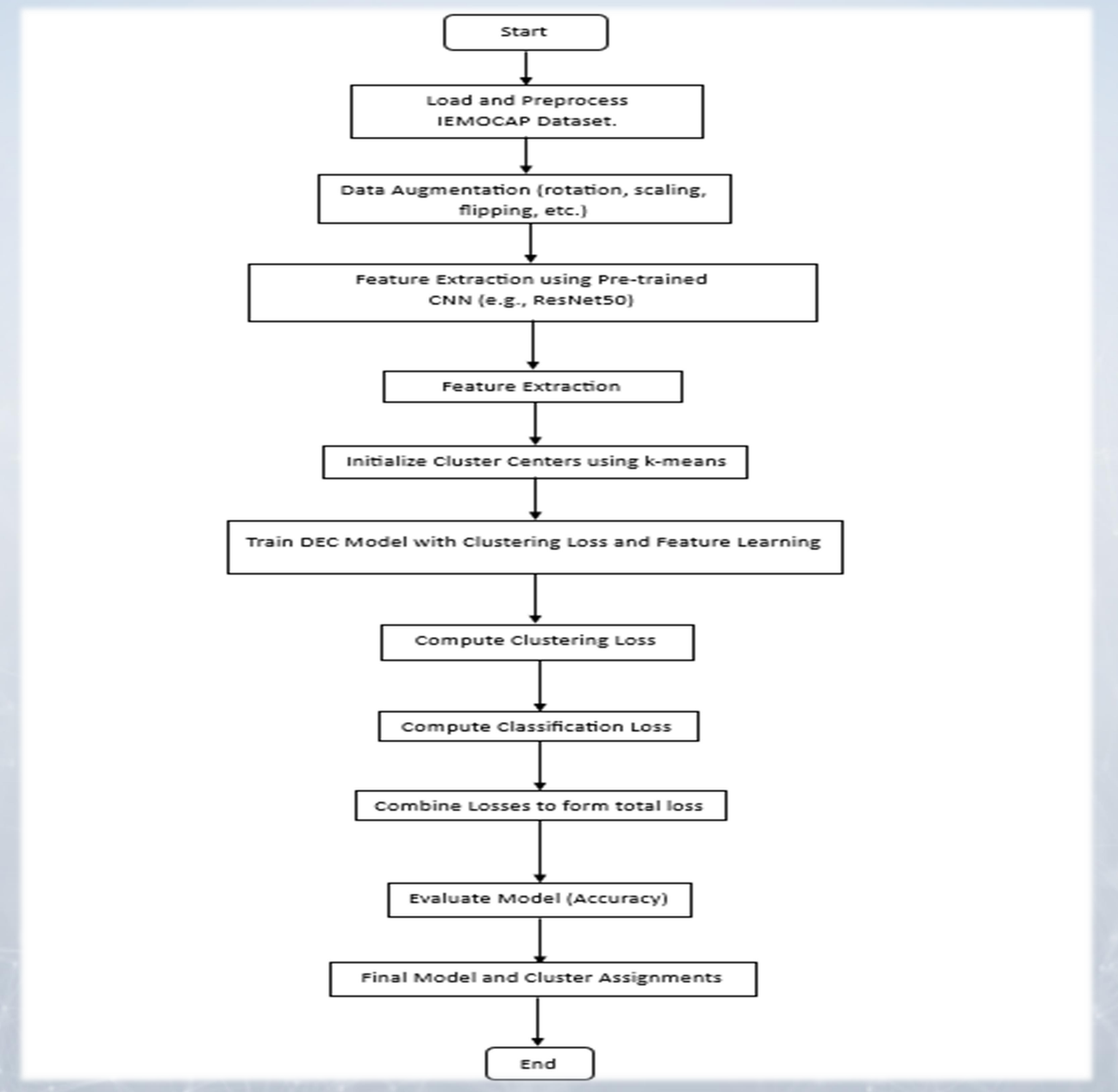


Fig. Flowchart of overall process

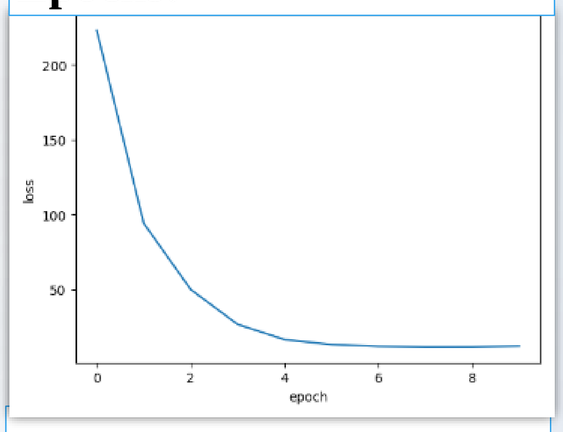
Tools:

- Keras
- TensorFlow
- Jupyter Notebook
- Scikit-Learn
- ResNet50V2

Training Progress & Loss Reduction:

```
Epoch 1/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 160s 26s/step - loss: 265.4775
Epoch 2/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 115s 27s/step - loss: 109.9543
Epoch 3/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 119s 27s/step - loss: 56.4500
Epoch 4/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 126s 29s/step - loss: 29.3725
Epoch 5/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 112s 26s/step - loss: 17.4437
Epoch 6/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 112s 26s/step - loss: 13.7889
Epoch 7/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 112s 26s/step - loss: 12.2977
Epoch 8/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 123s 29s/step - loss: 11.9467
Epoch 9/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 116s 27s/step - loss: 11.9815
Epoch 10/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 127s 29s/step - loss: 12.2163
```

Loss Reduction Over Epochs:



- **Initial Epochs:** The loss starts high (above 200) and drops significantly in the first few epochs. By the third epoch, the loss has decreased to below 50.
- **Mid Epochs:** Between epochs 3 and 6, the loss continues to decrease but at a slower rate, stabilizing around 20.
- **Later Epochs:** From epoch 7 to 10, the loss values show minimal change.

Conclusion:

- Implemented a semantic image clustering pipeline using ResNet50, contrastive learning, and K-Means clustering.
- Introduced custom clustering layers and consistency loss functions for automated large scale image dataset clustering.

SCAN ME