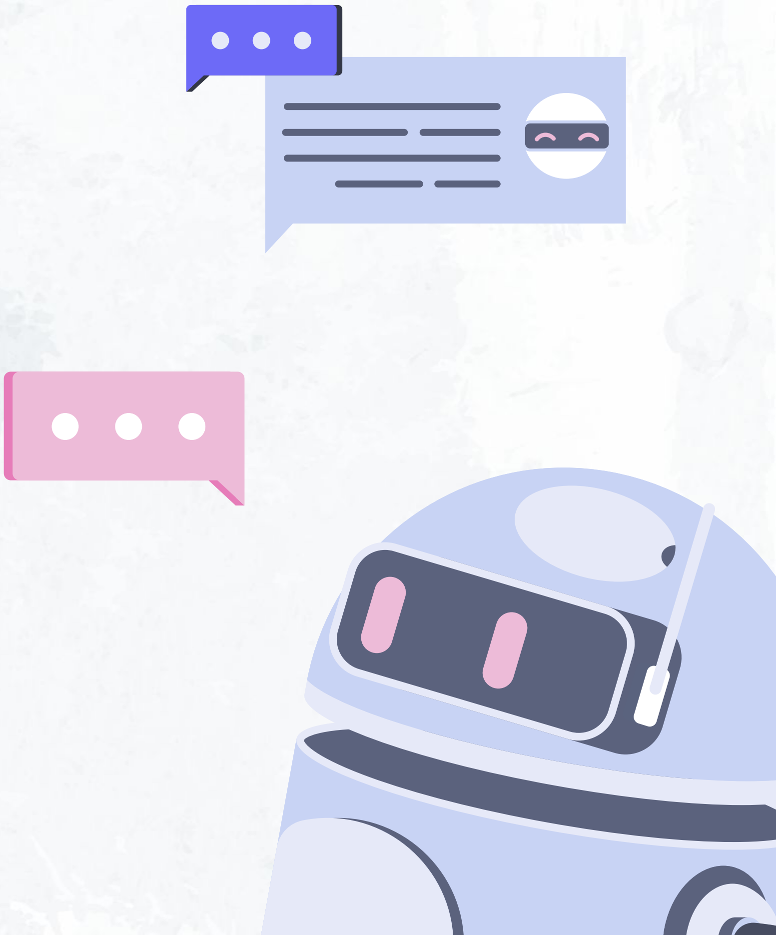
# Supervised Learning Exam

Università degli Studi di Milano-Bicocca

12-06-2023

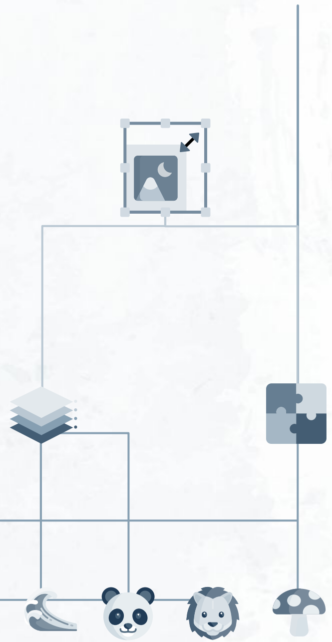


*Monaco Filippo – 840089*
*Picione Marco - 827116*

# Table of contents

00 →

# Introduction



**AIM:** Multi-class image classification
**MODELS:** Visual Bag of Words & Convolutional Neural Network

# Multi-class
# Image Classification

Computer vision task that aims to assign a single label to an image.

dog?

ant?

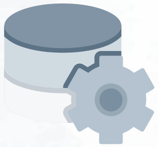
cat.

chair?

car?

01 →

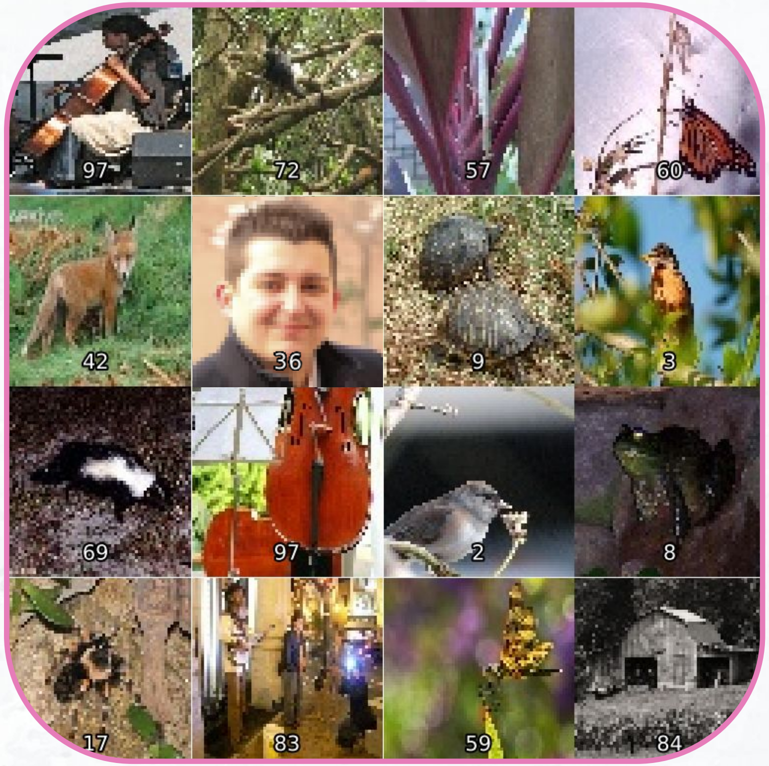
# Dataset



TinyImageNet, a reduced version of the widely-used ImageNet

- 100 classes
- 1000 images per class
- 64 x 64 image size

Data was split 80% for Training, 20% for Validation. A different set for Testing.




Some images of the dataset with associated label.

Note that the low resolution is due to the low size of the images (64x64).

# Data pre-processing

Bag of Words:

- RGB to GrayScale

Convolutional Neural Network:

- Resize to 256 x 256
- Central Crop to 224 x 224
- Image to Tensor
- Normalization (mean [0.485, 0.456, 0.406] , std [0.229, 0.224, 0.225])
- Batching (64 images per batch)

# Data Augmentation

Certain classes performed much worse than others.
Our Neural Network particularly did not enjoy pictures of cockroaches.

Augment this class by creating new images with random transformations:

- random Horizontal/Vertical Flip
- random Brightness variation
- random Rotation

ugly.

02 →

# Bag of Words

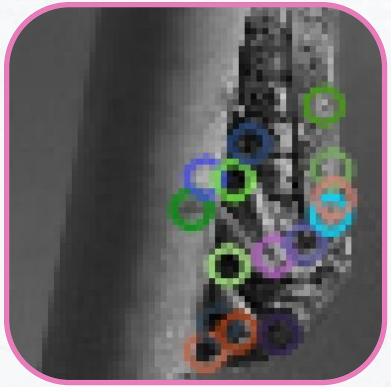
Visual Bag of Words tries to represent images by a set of features (keypoints and descriptors), which are used to construct a visual vocabulary.
The image is described by a frequency histogram of features, which is then used for classification.

# Extracting Local Features

SIFT is used to extract local features from images:

- keypoints (stand-out points)
- descriptors (description of keypoints)

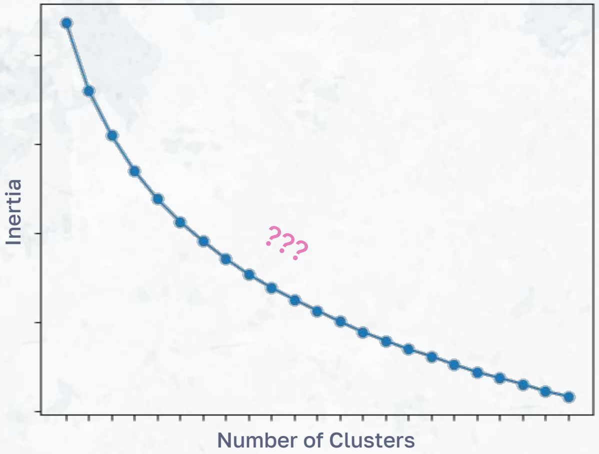
37 keypoints per image (on average), each described by a 128-dimensional vector.

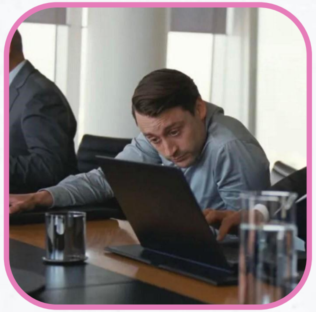


# Keypoints Clustering

Clustering of all descriptors using an unsupervised algorithm: MiniBatches k-Means.

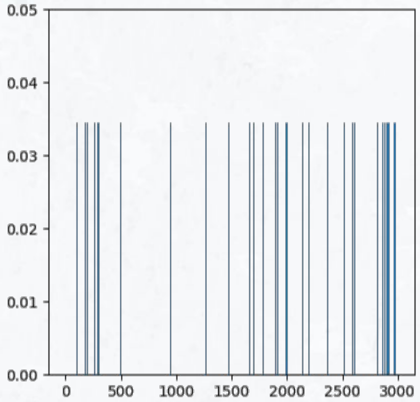
Elbow point method was inconclusive:
arbitrarily chosen 3000 clusters (number of visual words).



# Visual Histogram

Descriptors are clustered to construct a Visual Histogram based on the frequency of features w.r.t. the visual vocabulary.



# Classifier

Histograms are used to train a traditional classifier, which is then used to predict labels on the Test set.

Classifiers tested:

- k-Nearest Classifier
- Stochastic Gradient Descent Classifier
- Support Vector Classifier
- Decision Tree
- Random Forest
- AdaBoost
- Bagging

| | SVC | SGDC |
|---|---|---|
| Accuracy | 6.7 % | 4.7 % |
| Training Time | 4 hours | 4 minutes |
| Testing Time | 1 hour | 0.1 seconds |

# Motivation

Very poor performance achieved.

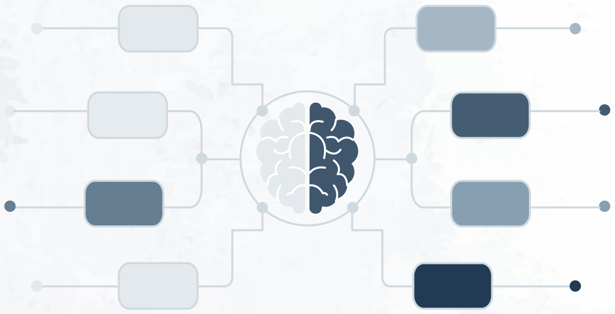
Possible reasons:

- The visual vocabulary is not optimal
- Overfitting

Further testing would be required.

03 →

# Convolutional Neural Network



Biologically-inspired Networks of neurons widely used in computer vision tasks. Stack multiple feature extractors as hidden neural layers and train them using ground truth labels.

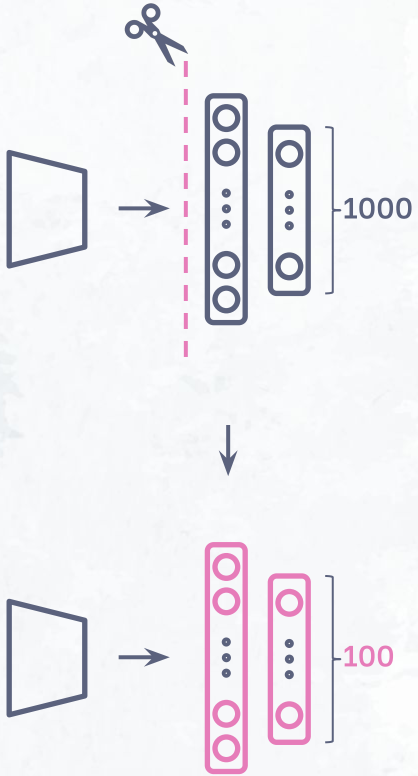
# Transfer Learning

1. Take pre-trained model
   - efficientnet_b0, trained on ImageNet

2. Change output classifier
   - from 1000 to 100 classes

3. Freeze main model
   - train only the newly-added classifier

4. Data pre-processing
   - Resize to 256 x 256
   - Central Crop to 224 x 224
   - Image to Tensor
   - Normalization (mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225])
   - Batching (64 images per batch)

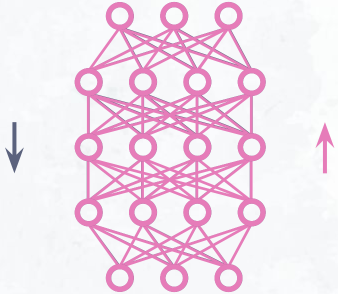


# Training and Validation

1. Input images to network with corresponding label
2. Output logits
3. Compute Cross-Entropy Loss
4. Backpropagate error, updating classifier parameters
5. Validate model at current epoch
6. Repeat process until validation loss is stable
7. Save best model

Hyper-parameters:

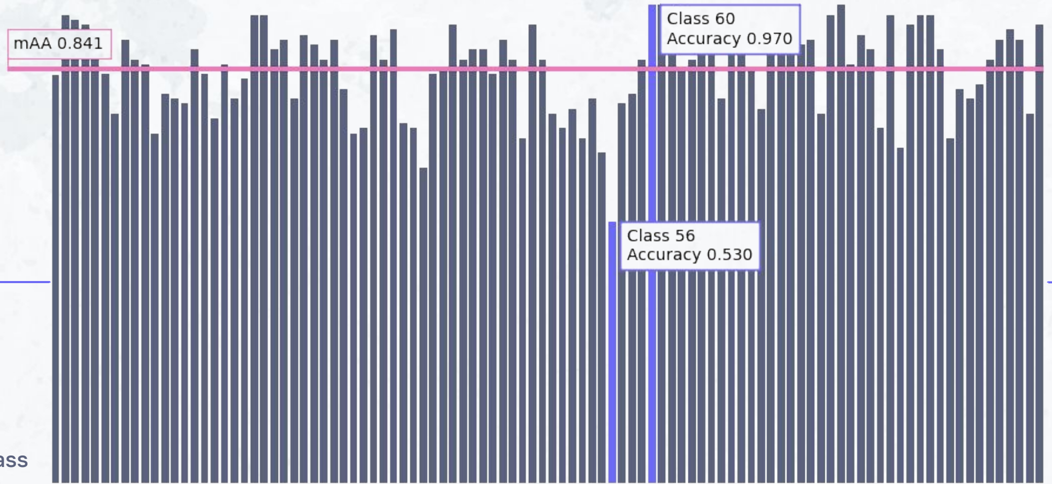
- Epochs : 10
- Patience : 3
- Optimizer : Adam
- Learning Rate : 0.001
- Scheduler : CosineAnnealingLR



# Testing

Metrics:
- Accuracy per class
- mean Average Accuracy: 84.1%

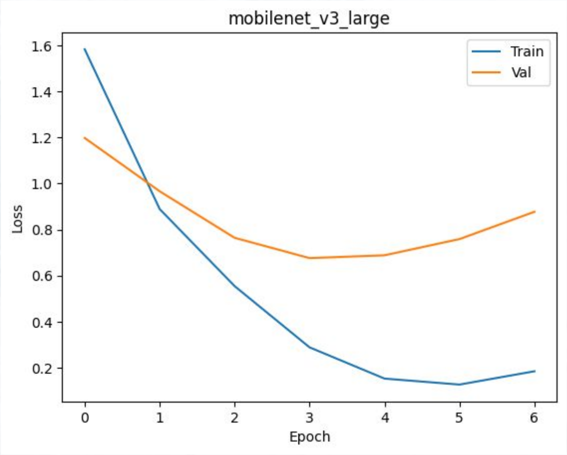
Data augmentation was successful on class 56: from 53.0% to 75.0% accuracy.

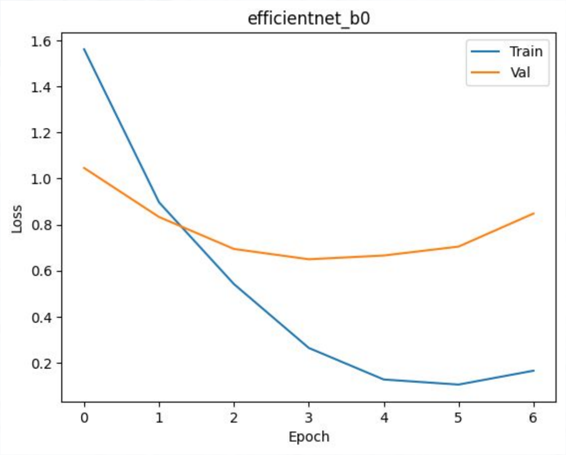



Accuracy per Class

# Hyperparameter Tuning

To achieve the best possible performance, we tried many different pre-trained models:
- mobilenet_v3_large 79% accuracy
- efficientnet_b0 84% accuracy
- vgg16_bn 70% accuracy



# Optuna

Python framework for automatic hyperparameter optimization.
Sample different sets of values for each parameter and train the network, try to minimize the loss in validation.

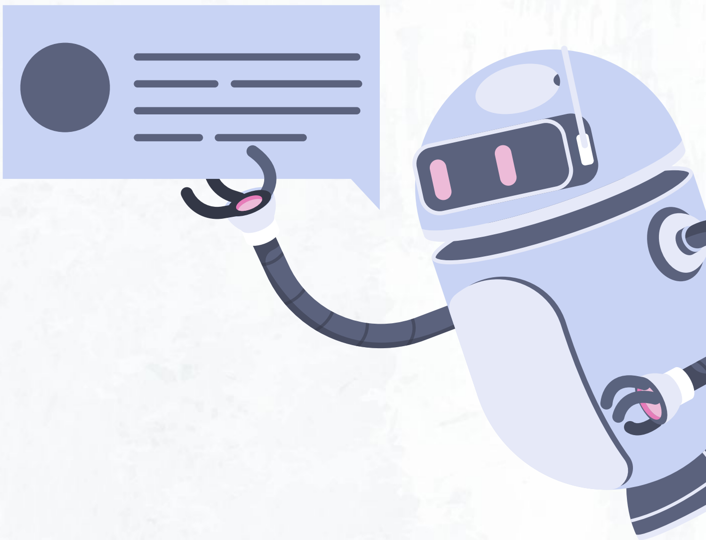
Hyperparameters:

- Number of Hidden Neurons in Classifier (first and second hidden layer)
- Dropout chance
- Training Epochs
- Learning Rate
- Optimizer

Colab did not want us to reach the maximum of our abilities.

04 →

# Conclusion



The Convolutional Neural network works much much better than the traditional classifier, both in terms of overall accuracy, and training times.

And them’s the facts.

# Thanks!

If you have any more questions, please direct them to
chat.openai.com
(We did not make us of ChatGPT or any other Natural Language Processing models for this project).


Credit:
Filippo Monaco & Marco Picione
under the wise supervision of Prof. Simone Bianco & Mirko Agarla