Università degli Studi di Milano – Bicocca
Corso di Laurea Magistrale in Data Science
Anno Accademico 2022/2023

Digital Signal and Image Management Project

Signal and Image Processing
Content based Image Retrieval

Julia Lan Bui Xuan
Silvia Grosso

Music Genre Classification

GTZAN Dataset

- **1000 audio** tracks, each **30 seconds** long
- **10 genres**, each represented by 100 tracks
- Genres:
    1. Blues
    2. Classical
    3. Country
    4. Disco
    5. Hiphop
    6. Jazz
    7. Metal
    8. Pop
    9. Reggae
    10. Rock

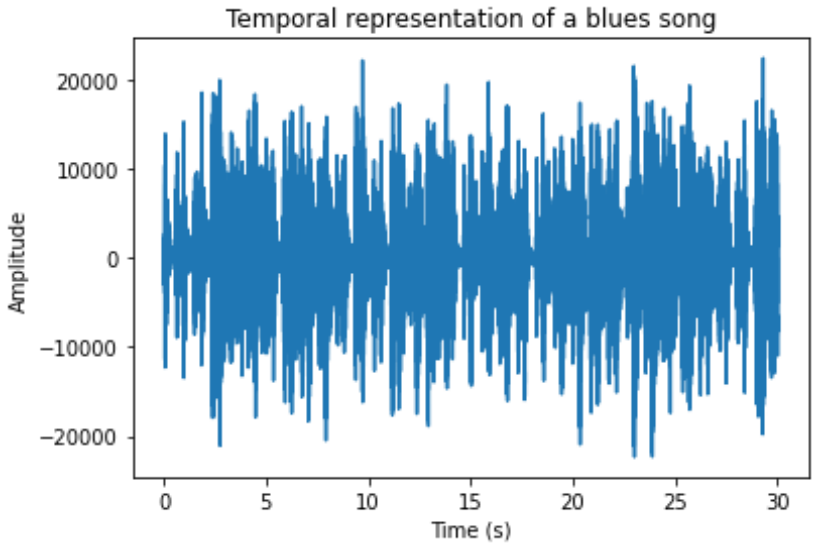
Data preprocessing

- One file removed because of the incorrect format

- **Train** set = 719 files
- **Validation** set = 180 files
- **Test** set = 100 files

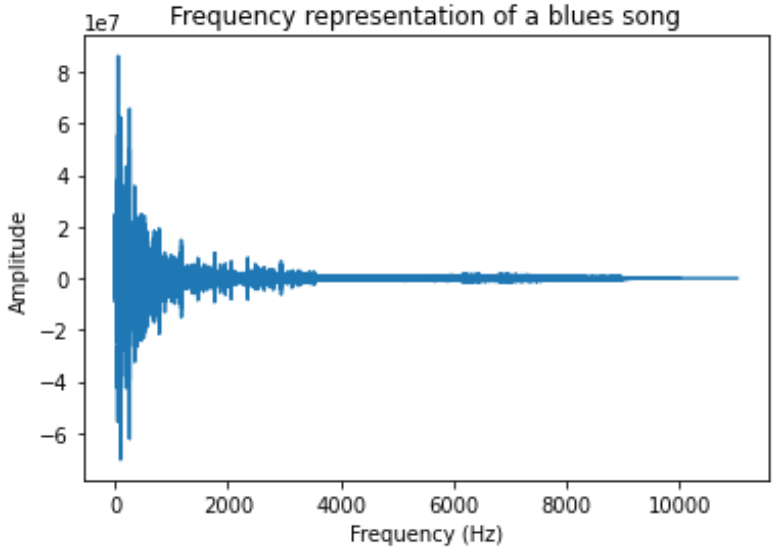
90% → 80%, 20%

10%



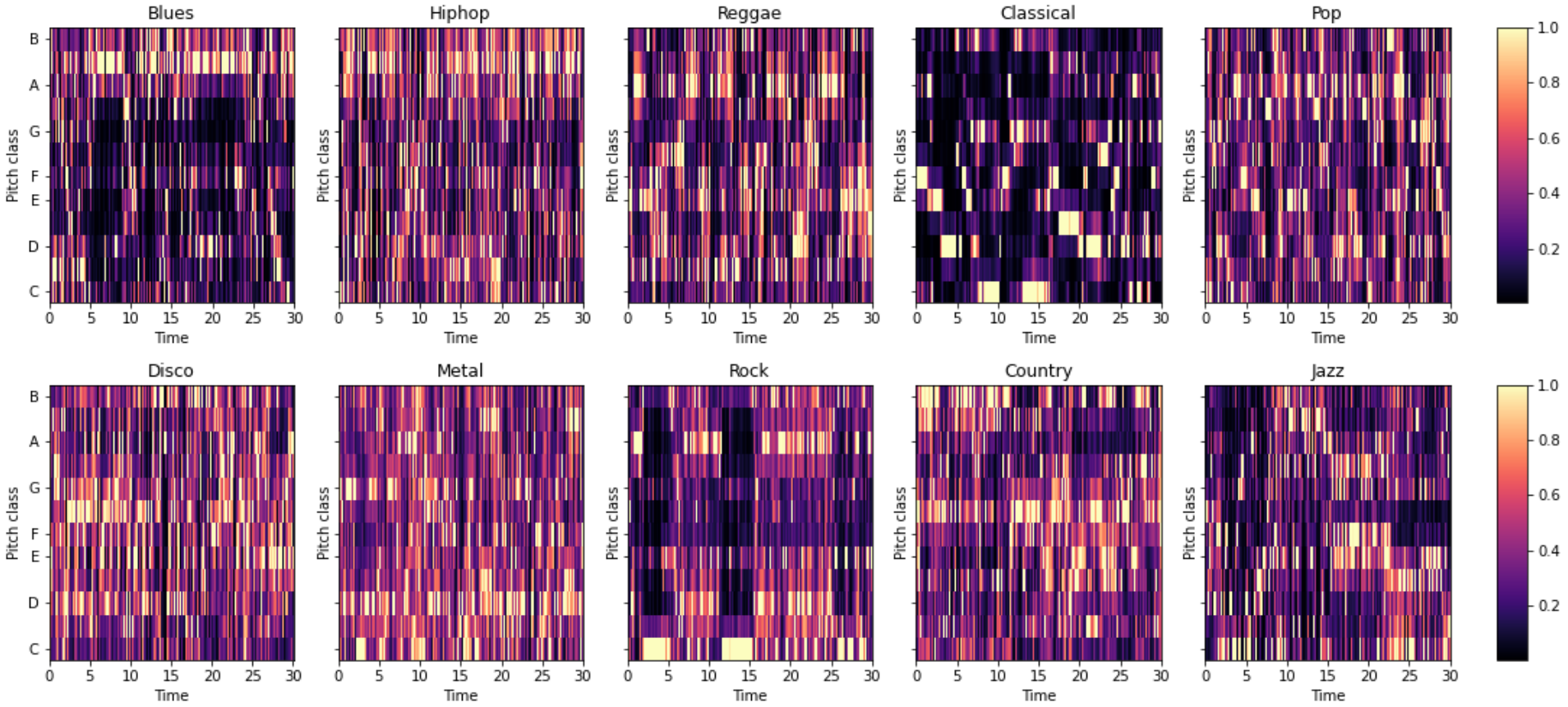
Raw audio data: one-dimensional signal, with thousands of samples (in our case 661794 samples per song)

Extracting **two-dimensional features** provides a compact and interpretable representation of the data, while also improving computational efficiency and robustness



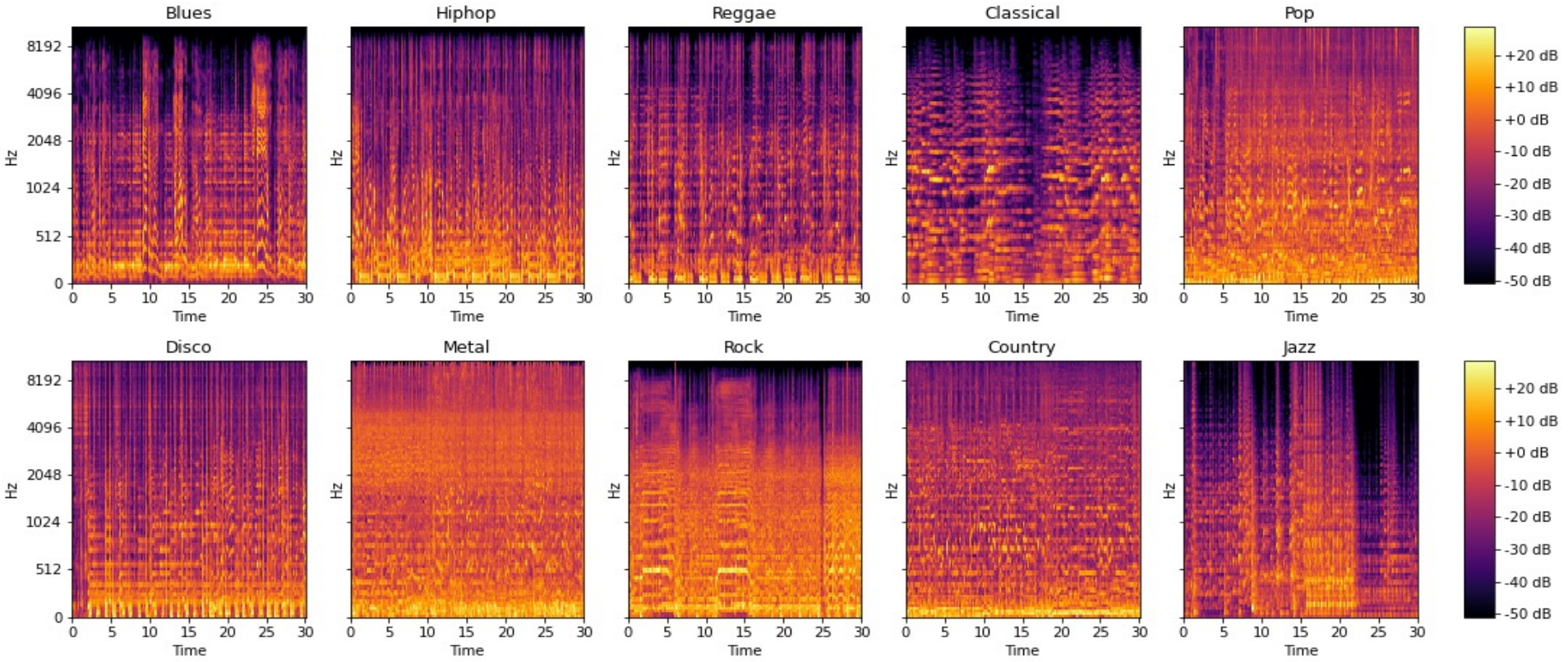
Feature extraction: Chroma features

- Shape: (12, 1293)
- Captures harmonic and melodic characteristics, primarily the 12 pitch classes



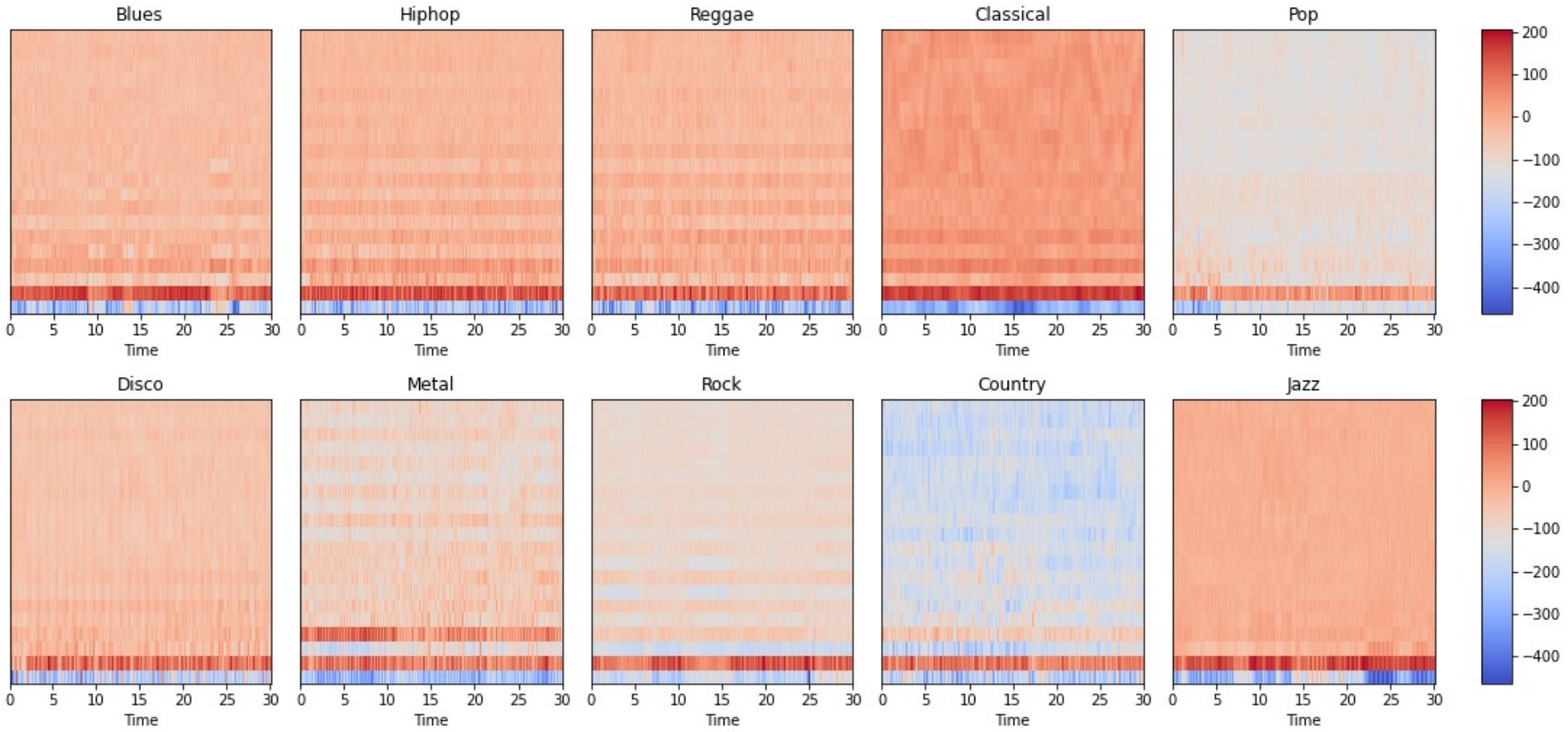
Feature extraction: Mel spectrograms

- Shape: (128, 1293)
- Spectrum of frequencies of an audio signal over time
- Frequencies converted to the Mel Scale



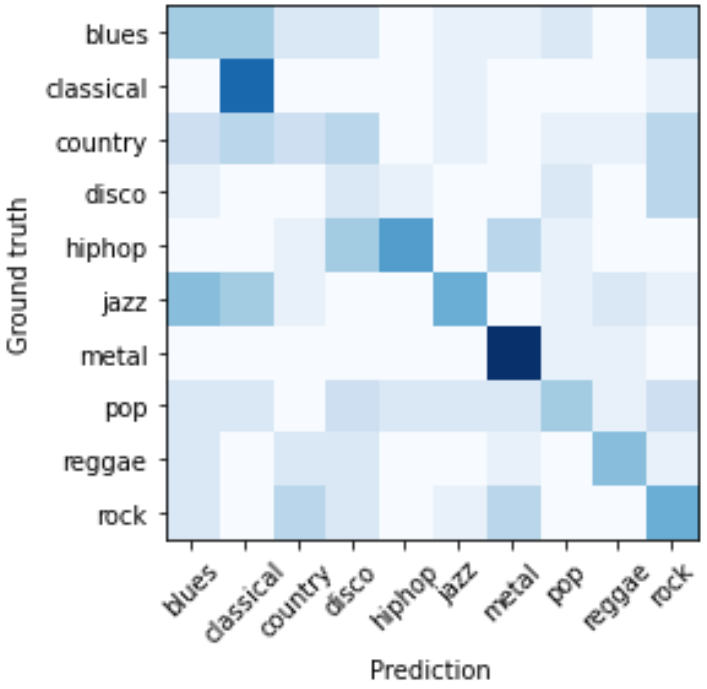
Feature extraction: MFCC spectrograms

- Shape (20, 1293)
- Derived from the Mel spectrogram; coefficients capture the spectral envelope of the sound signal and provide a compact representation of its spectral content.



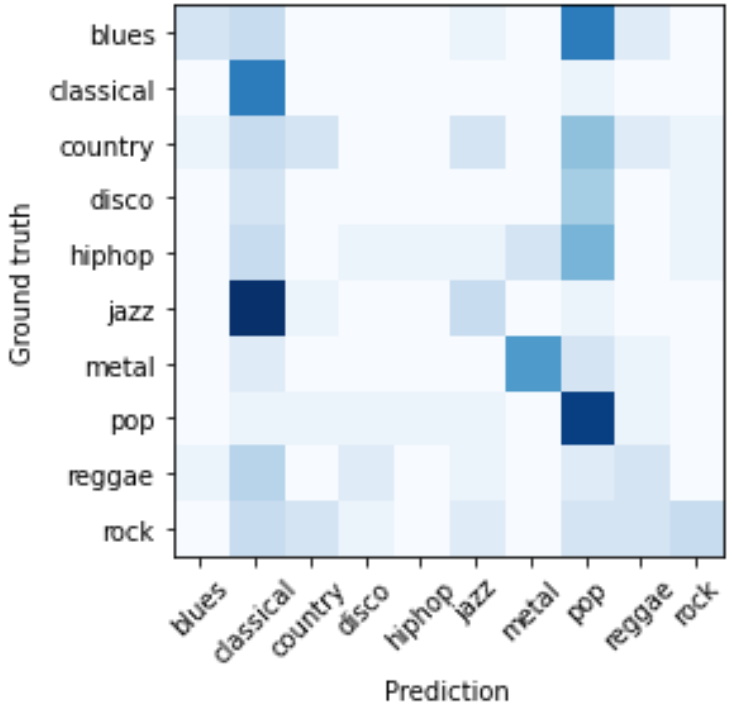
SVM - Results

Chroma features



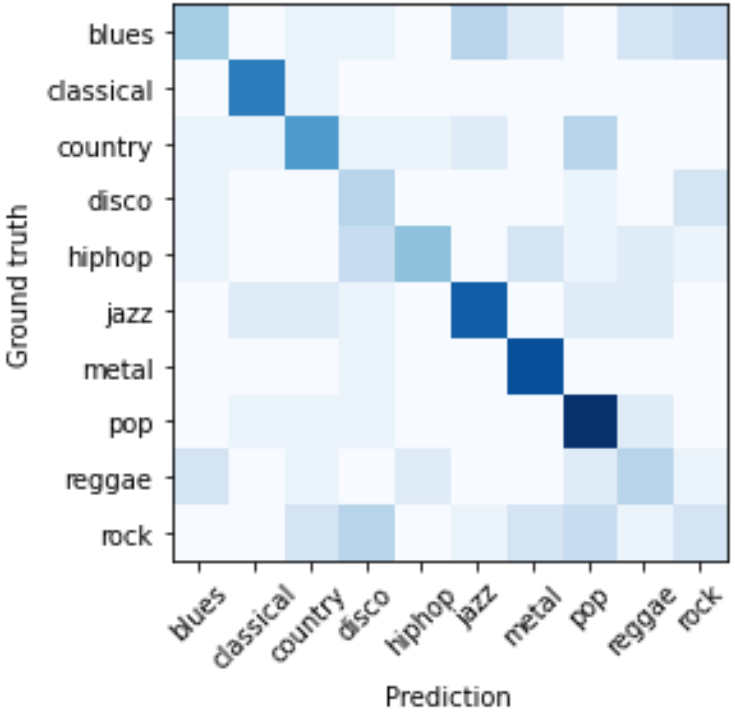
Accuracy: 38%

Mel spectrogram



Accuracy: 31%

MFCC spectrogram



Accuracy: 52%

- SVM with **MFCC spectrograms** extracted as features reached the highest accuracy
- Therefore, they will be used as **input in the Convolutional Neural Networks**

Data augmentation

Dataset – further preprocessing

How long does it take to listen to a song before determining its genre?

30 seconds are little too much information

- ❑ A **5-sec** version of the dataset was created, helping to mitigate the small dataset size
- ❑ Generated by dividing each 30-sec song into 6 segments
- ❑ **The corresponding MFCC spectrograms** were extracted from the 5-sec audio clips

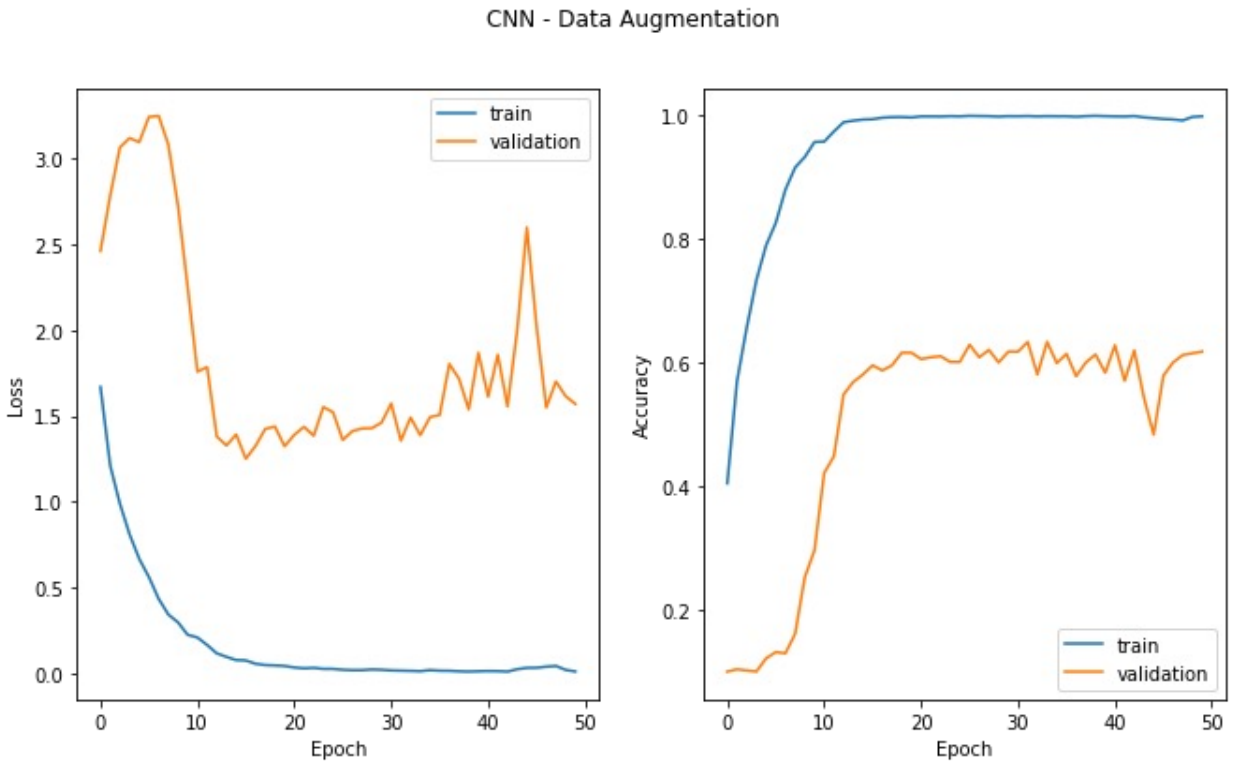
→ *6x increase*
Train set: 4314
Validation set: 1080

Audio signals - augmentation

- ❑ artificially creating new training samples
- ❑ **White noise addition**: involves adding random Gaussian noise to the original audio
- ❑ Noise factor = 0.3
- ❑ Applied to **30% of training audio files**

→ Train set: 5609

CNN 2



- **Accuracy: 62%**
- **Top 3 accuracy: 87%**

Architecture

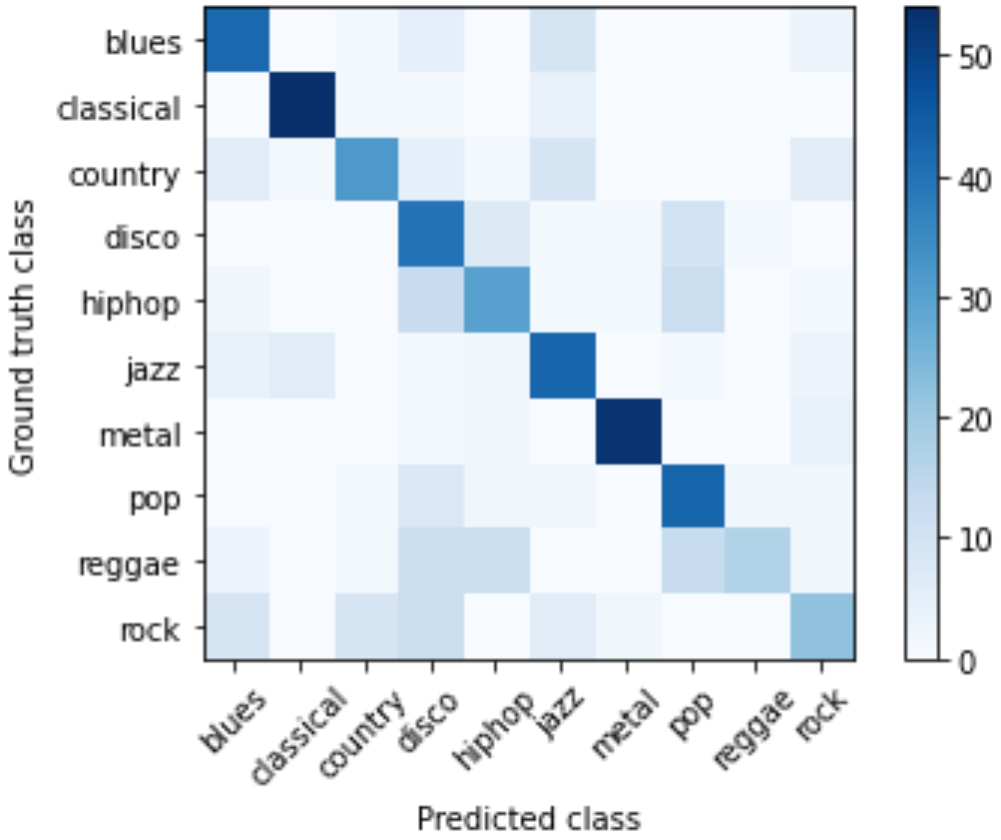
- Input layer with shape (224,224,3)
- 5 blocks of:
  - Convolutional, Batch Normalization and MaxPooling layers, which extract features from the input image
- ReLU activation
- Fully connected (dense) layer with **softmax activation**, which outputs the class probabilities
- Increasing number of filters in each layer, 8, 16, 32, 64, and 128.

Test set - evaluation

- Accuracy = 63%
- Top 3 Accuracy = 88%

The most correctly predicted class: classical, metal

The least correctly predicted class: reggae, rock



Test set - evaluation

Example. Top 3 genres prediction of a 30 seconds blues song, **on all of its 5 seconds segments**

Probability	Genre
0.733713	blues
0.118467	hiphop
0.090324	country

segment 1

Probability	Genre
0.607904	disco
0.232906	blues
0.085626	country

segment 2

Probability	Genre
0.967822	blues
0.016397	rock
0.011934	country

segment 3

Probability	Genre
0.382395	blues
0.318152	hiphop
0.104593	country

segment 4

Probability	Genre
0.930740	disco
0.053135	blues
0.008810	country

segment 5

Probability	Genre
0.336893	country
0.200564	blues
0.178876	rock

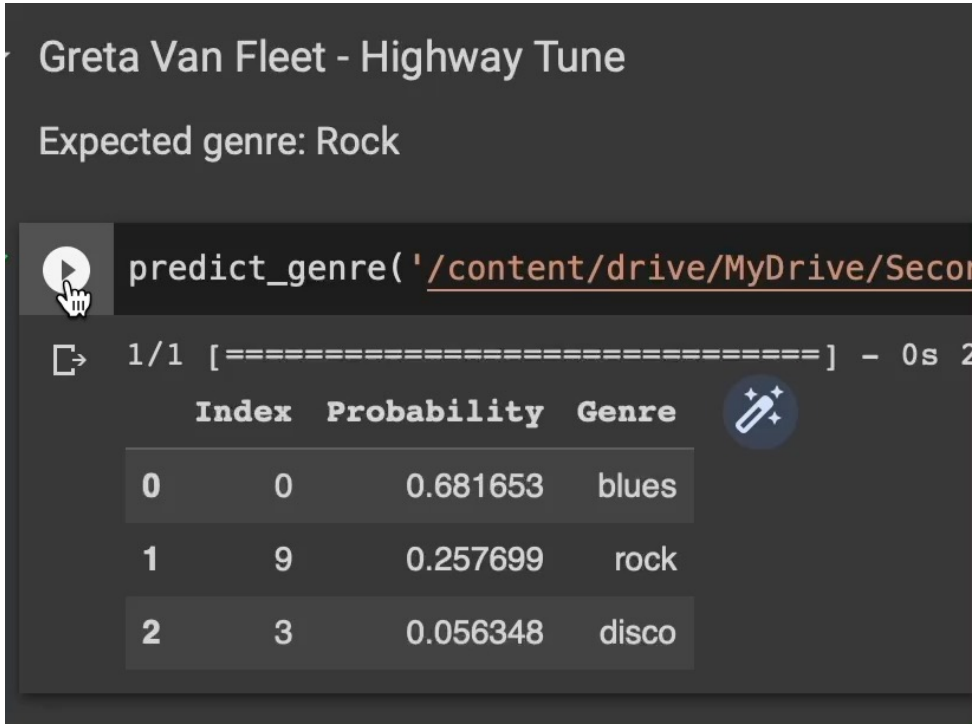
segment 6

Top 1 accuracy: 50%

Top 2 accuracy: 100%

Demo

Song out of the test set – downloaded from internet



Top 3 genres prediction, based on **5 second segment randomly extracted** of a rock song.

Most of the times, the expected rock genre is in the top 3 predictions.

Food Classification

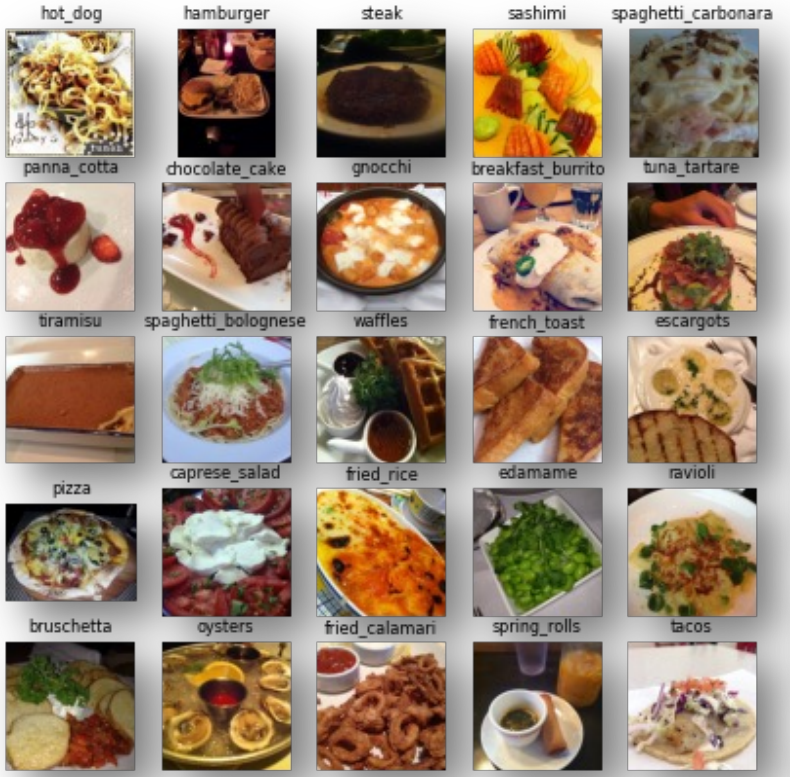
FOOD-101 Dataset

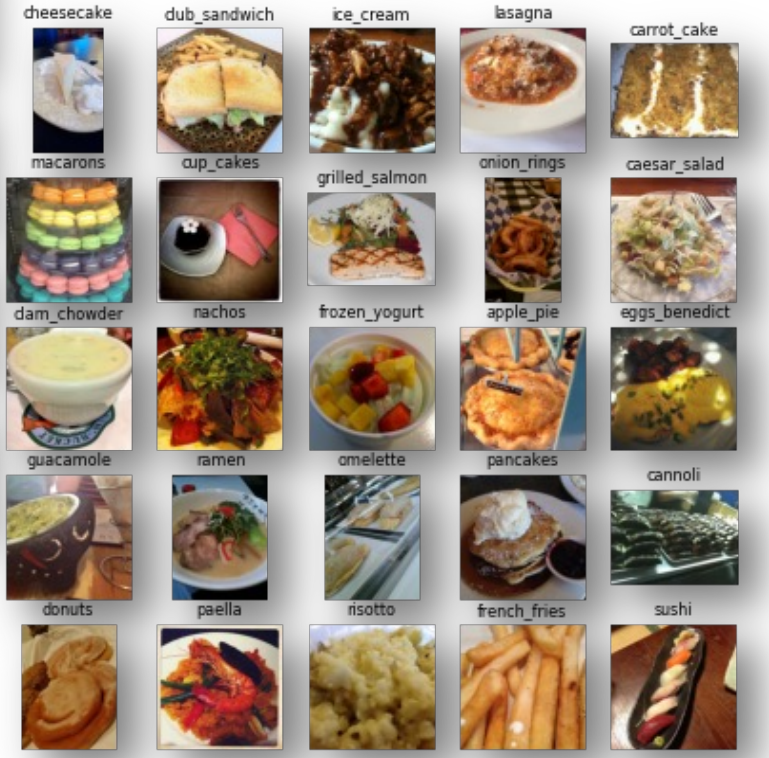
Random sample to **50 classes**

- **70%** training set
- **30%** test set

Training set splitted into

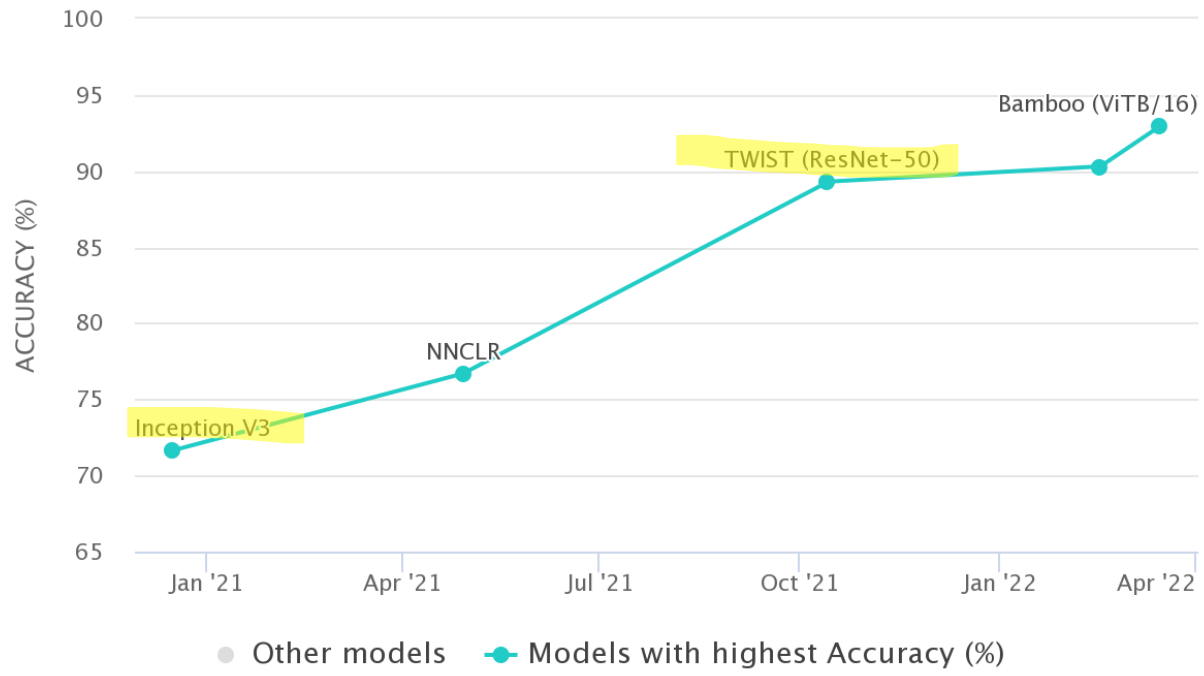
  - **80%** train
  - **20%** validation





- ❑ Different **framing**
- ❑ Different **brightness**
- ❑ Presence of **wrong images**

Transfer Learning



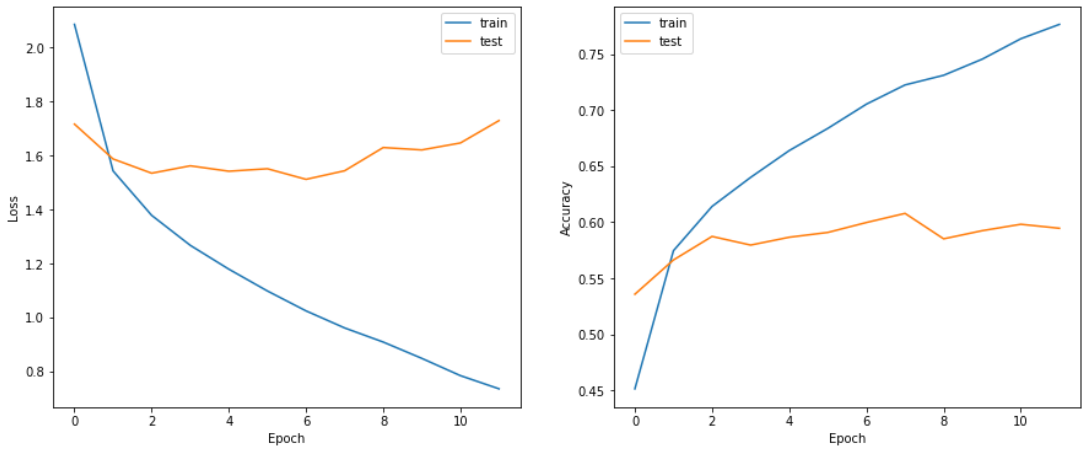
ResNet50
vs
InceptionV3

Transfer Learning – InceptionV3

Data Augmentation

- **RandomFlip** Horizontal and Vertical
- **RandomContrast(0.25)**: between 75 and 125 % of the original values

Loss and Accuracy on train and validation set

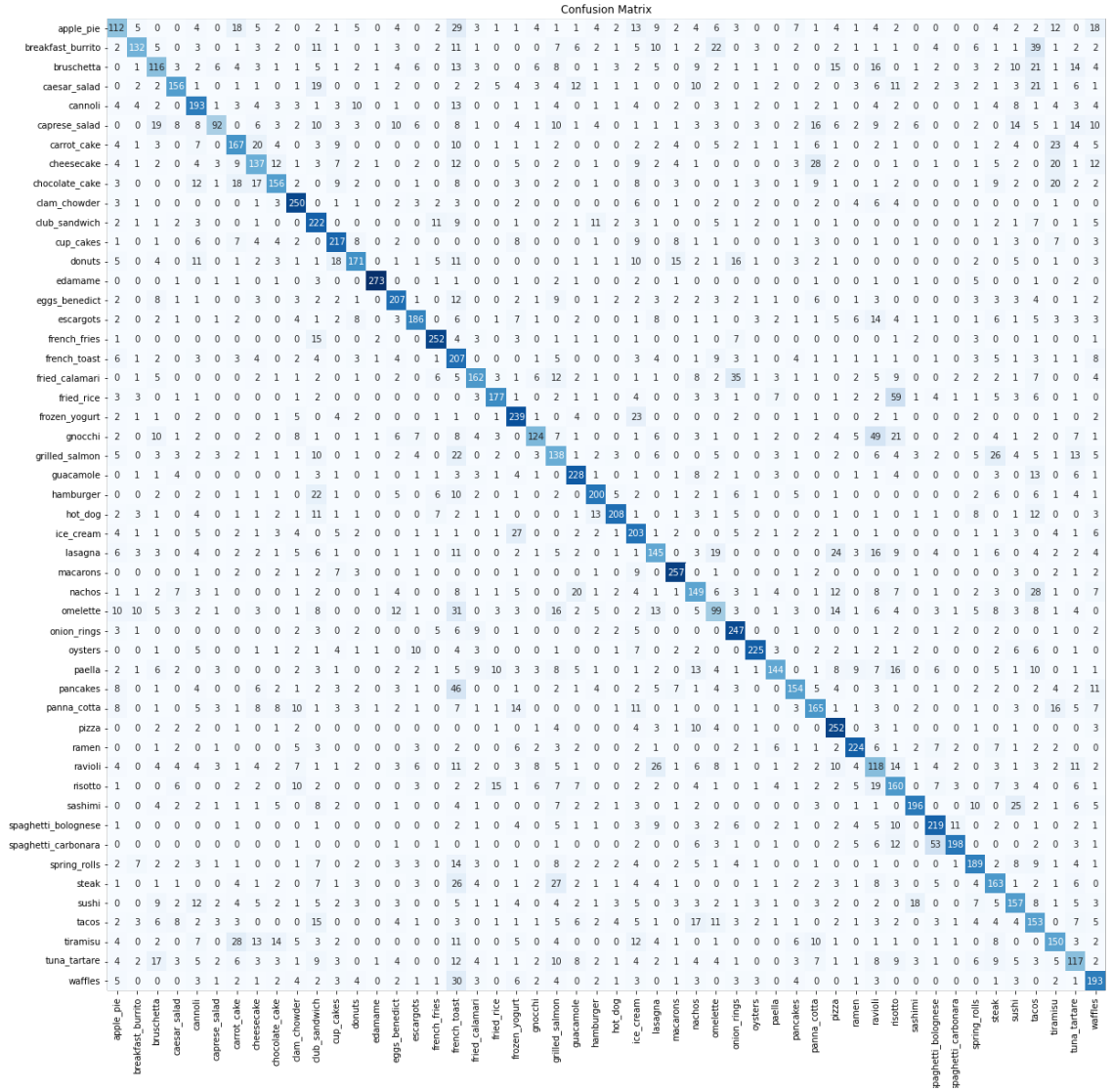


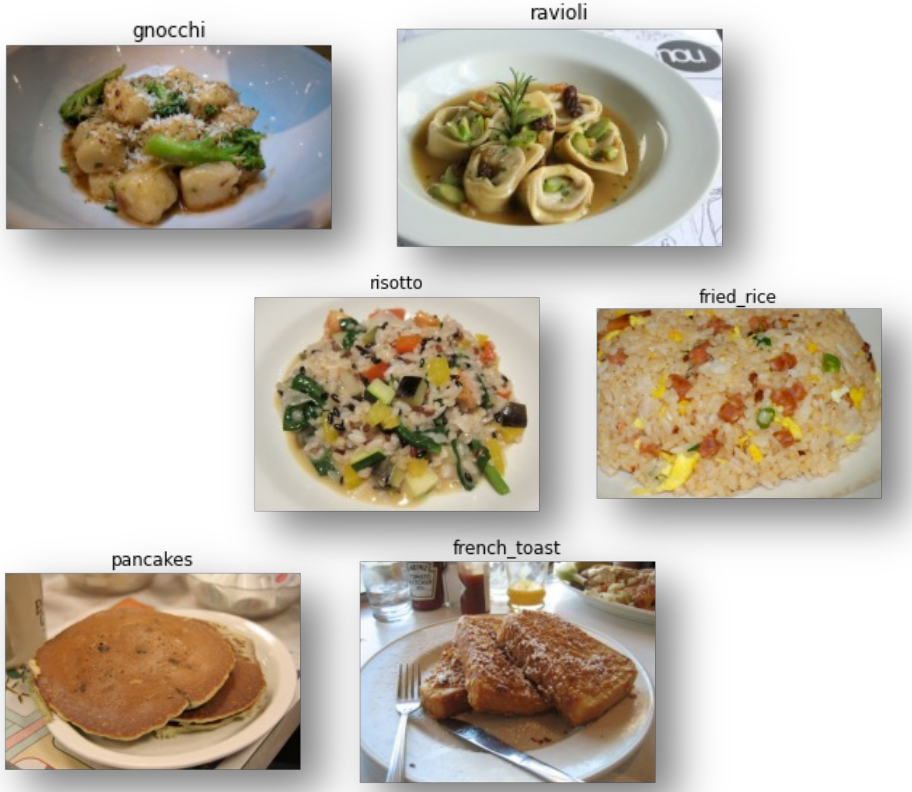
Final Architecture

- Input shape (224,224,3)
- **InceptionV3** freezing layers
- Fully connected **Dense** layer
- **ReLU** activation
- **Dropout** (0.2)
- Final **softmax** activation

- Optimizer **Adam**
- Learning rate **0.001**
- **Categorical crossentropy** loss

Prediction of the best net

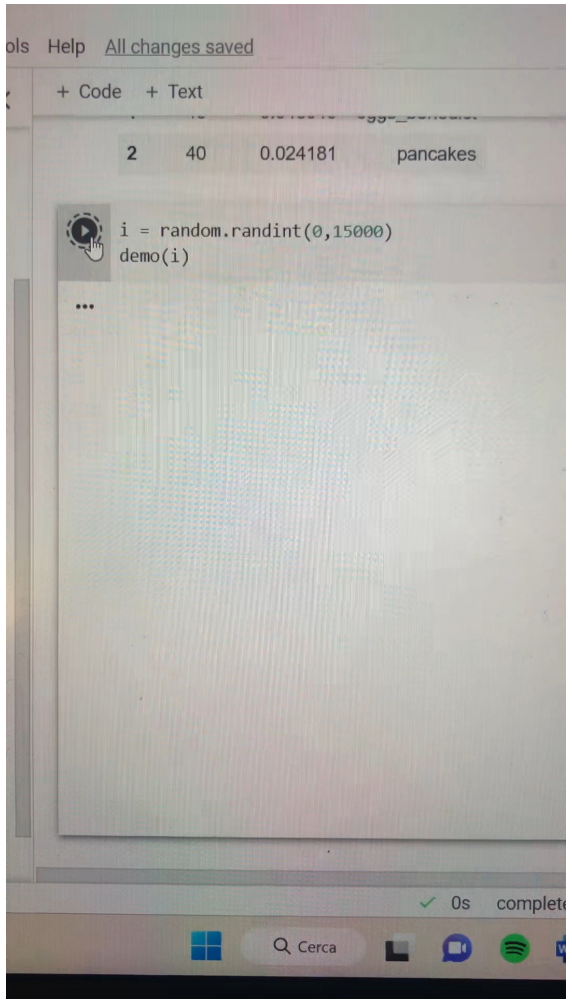


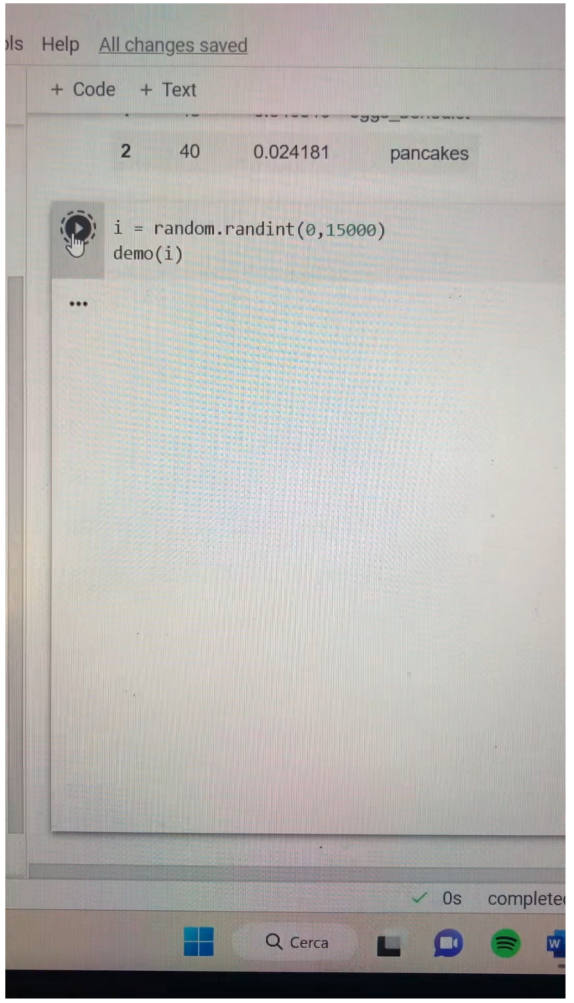


Top 1 Accuracy : 60%

Top 3 Accuracy : 79%

Demo

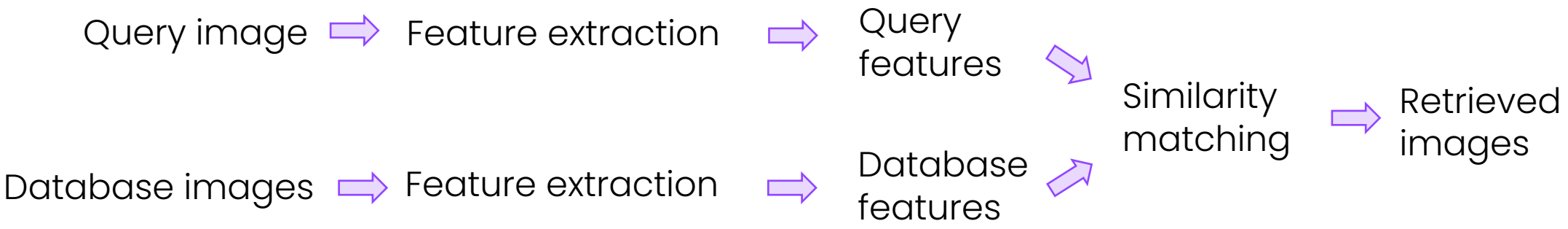

Help All changes saved
+ Code + Text
2 40 0.024181 pancakes
i = random.randint(0,15000)
demo(i)
0s
Cerca


Help All changes saved
+ Code + Text
2 40 0.024181 pancakes
i = random.randint(0,15000)
demo(i)
0s
Cerca

Content Based Image Retrieval

FOOD 101 dataset – 50,000 images

- **Database** images : 90%
- **Query** images : 10%



- Neural features: extracted from a CNN based on the Residual Network with 50 layers architecture (**ResNet50**)
- Similarity matching: **KDTREE**

Quantitative evaluations

k-Precision

For each class *c*: $P_c(k) = Q_{kc}/Q_c$, where

- k = number of images to retrieve
- Q_{kc} = correct queries among the first *k* retrieved images in the class *c*
- Q_c= number of queries (test images) from class *c* (=100)

$$P_c(1) = \frac{Q_{1c}}{Q_c} = \frac{1}{100}\sum_{i=1}^{100} x_i$$

$$P_c(10) = \frac{Q_{10c}}{Q_c} = \frac{1}{100}\sum_{i=1}^{100} \frac{x_i}{10} \qquad x_i \in \{0,1\}$$

$$P_c(100) = \frac{Q_{100c}}{Q_c} = \frac{1}{100}\sum_{i=1}^{100} \frac{x_i}{100}$$

Average 1-precision: $P(1) = \frac{\sum_{c=1}^{50} P_c(1)}{50} = 50.12\ \%$

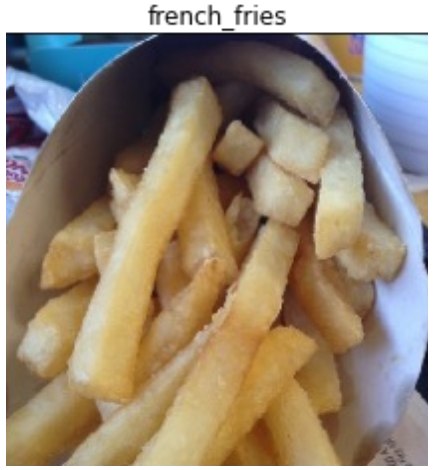
Average 10-precision: $P(10) = \frac{\sum_{c=1}^{50} P_c(10)}{50} = 41\ \%$

Steak: $Min\ P_c(1) = 22\%$

Edamame: $Max\ P_c(1) = 95\%$

Evaluation : TEST 1

Query image



$$P_c(10) = 100\%$$

$$\overline{P_c(1)} = 80\%$$

10 most similar images retrieved

Evaluation : TEST 2

Query image

fried_calamari

$$P_c(10) = 90\%$$

$$\overline{P_c(1)} = 43\%$$

10 most similar images retrieved

1: fried_calamari

2: fried_calamari

3: fried_calamari

4: fried_calamari

5: fried_calamari

6: fried_calamari

7: fried_calamari

8: fried_calamari

9: onion_rings

10: fried_calamari

Evaluation : TEST 3

Query image – *out of the test set*

frozen_yogurt

$$P_c(10) = 40\%$$

$$\overline{P_c(1)} = 58\%$$

10 most similar images retrieved

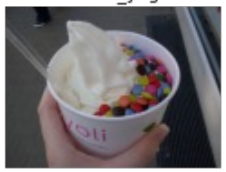
1: ice_cream

2: frozen_yogurt

3: panna_cotta

4: frozen_yogurt

5: frozen_yogurt



6: ice_cream

7: breakfast_burrito

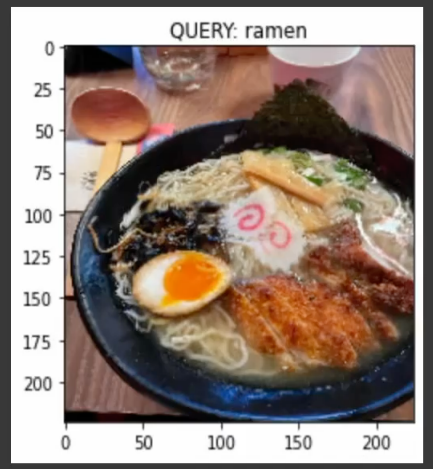
8: frozen_yogurt

9: oysters

10: panna_cotta

Demo Image Retrieval

```
query_image = kimage.load_img('/content/gdrive/MyDrive/Second year/DSIM_project/ramen_milano.jpg', target_size=(224, 224))
query_label = 'QUERY: ramen'
plt.imshow(query_image)
plt.title(query_label)
plt.show()
```



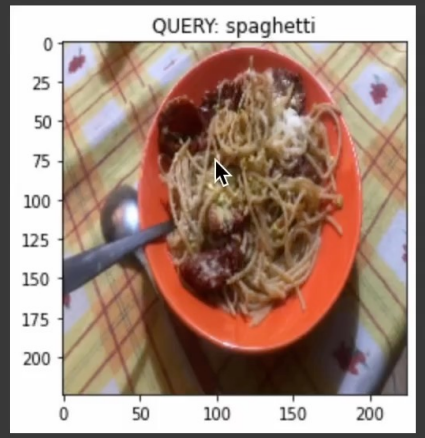

```
[22] # Computing query features
query_features = neural_features(query_image)
# Adding one dimension as required by the KDTree
query_features = np.expand_dims(query_features, axis=0)
# Search, k = 100
dist, ind = tree.query(query_features, k=100)

plot_10(query_image)
```

Demo Image Retrieval

+ Code + Text

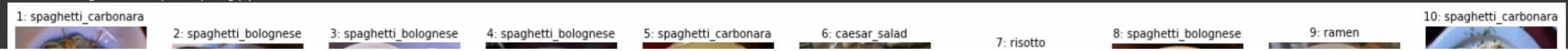
RAM Disk

```
query_image = kimage.load_img('/content/gdrive/MyDrive/Second year/DSIM_project/spaghetto.jpg', target_size=(224, 224))
query_label = 'QUERY: spaghetti'
plt.imshow(query_image)
plt.title(query_label)
plt.show()
```




```
[20] # Computing query features
     query_features = neural_features(query_image)
     # Adding one dimension as required by the KDTree
     query_features = np.expand_dims(query_features, axis=0)
     # Search, k = 100
     dist, ind = tree.query(query_features, k=100)

     plot_10(query_image)
```

```
/usr/local/lib/python3.8/dist-packages/PIL/TiffImagePlugin.py:788: UserWarning: Corrupt EXIF data.  Expecting to read 12 bytes but only got 0.
  warnings.warn(str(msg))
```



Possible future developments

Processing mono-dimensional signals

- ❑ Dataset containing more genres – subgenres
- ❑ Data augmentation, not only on audio data, but also on the extracted bidimensional features

Processing bi-dimensional signals

- ❑ Gamma correction on the images
- ❑ Fine Tuning of InceptionV3

Image Retrieval

- ❑ Try with other space-partitioning data structures
- ❑ Try query expansion techniques

References

- Gianluigi Ciocca, Paolo Napoletano, Raimondo Schettini, CNN-based features for retrieval and classification of food images, Computer Vision and Image Understanding, Volumes 176–177, 2018, Pages 70-77, ISSN 1077-3142, https://doi.org/10.1016/j.cviu.2018.09.001
- Hendrik Purwins∗, Bo Li∗, Tuomas Virtanen∗, Jan Schluter∗, Shuo-yiin Chang, Tara Sainath, Deep Learning for Audio Signal Processing, JOURNAL OF SELECTED TOPICS OF SIGNAL PROCESSING, VOL. 13, NO. 2, MAY 2019, PP. 206–219, http://doi.org/10.1109/JSTSP.2019.2908700
- Papers With Code, Image Classification on Food-101, https://paperswithcode.com/sota/image-classification-on-food-101-1
- Lorenzo Famiglini, Transfer Learning with Deep Learning & Machine Learning techniques https://medium.com/@lorenzofamiglini/transfer-learning-with-deep-learning-machine-learning-techniques-b4052befe7e2
- G.Tzanetakis and P. Cook, 2002, GTZAN dataset, https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification