

# **Digital Signal and Image Management Project**

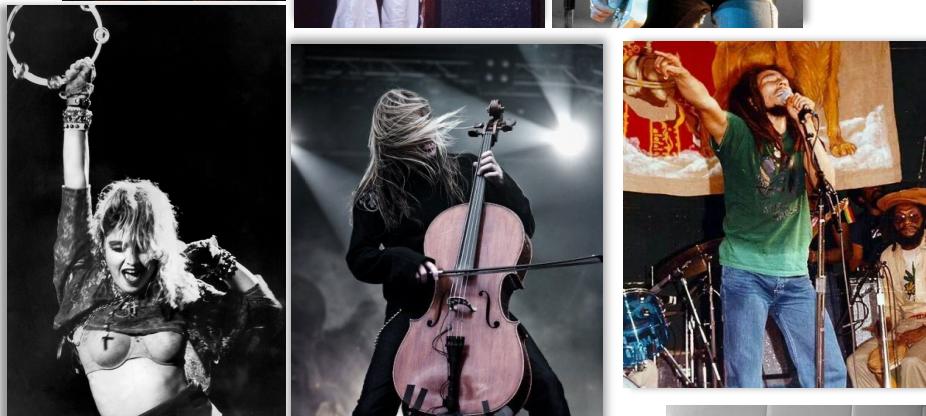
**Signal and Image Processing  
Content based Image Retrieval**

Julia Lan Bui Xuan  
Silvia Grosso

# Music Genre Classification

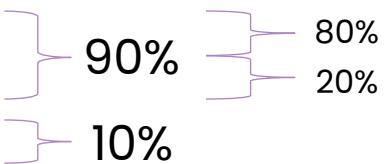
## GTZAN Dataset

- ❑ 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



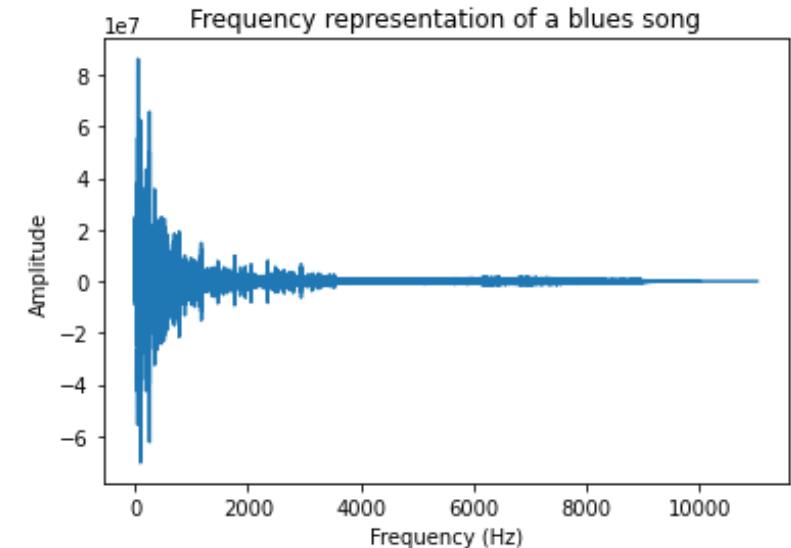
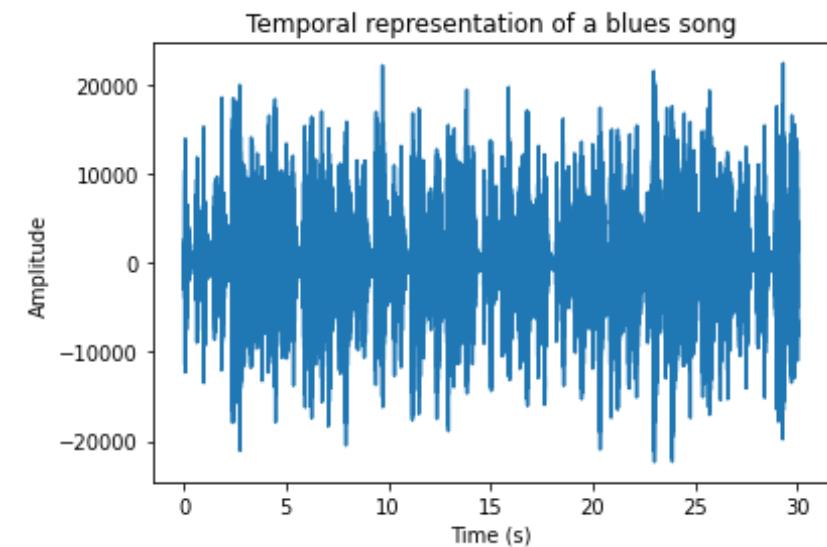
# Data preprocessing

- ❑ One file removed because of the incorrect format

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

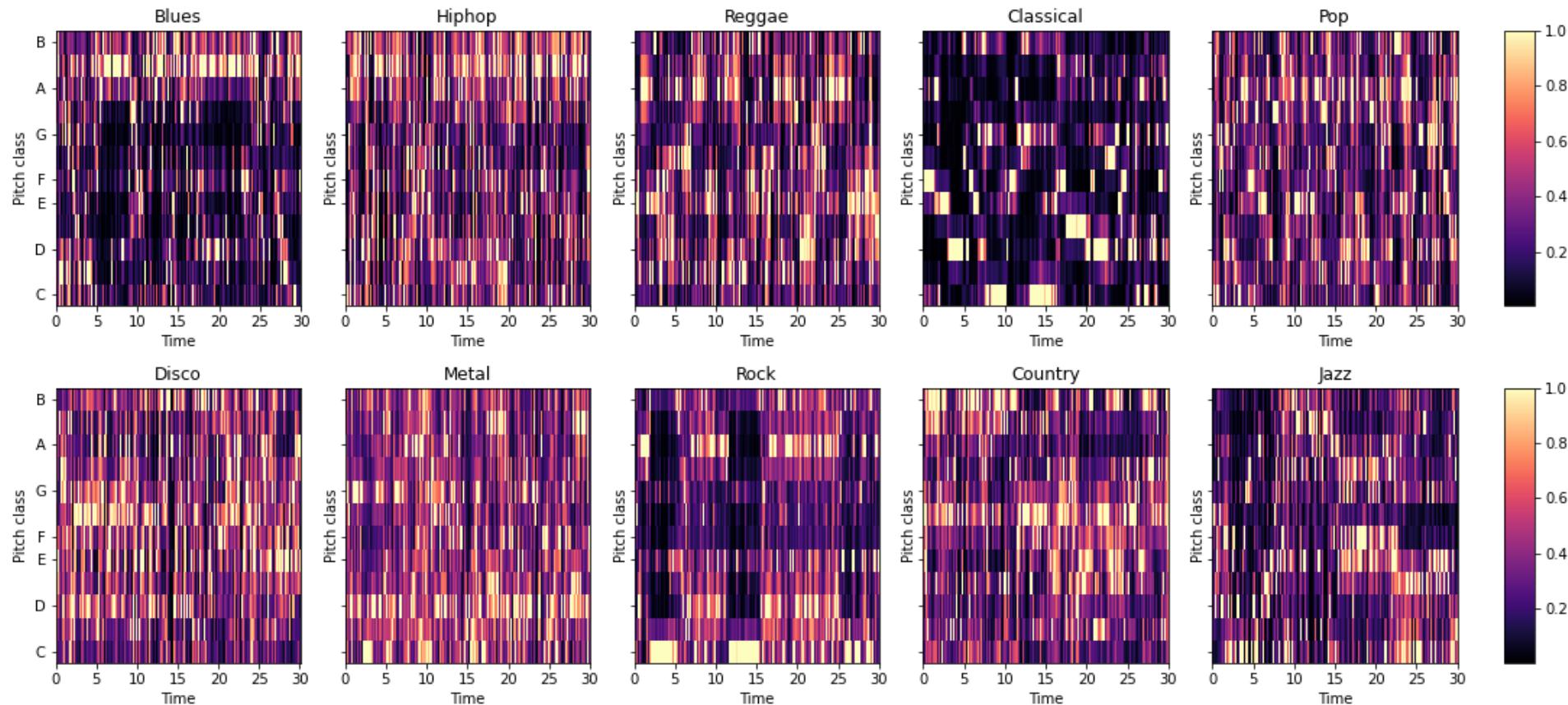
**Raw audio data:** one-dimensional signal, with thousands of samples (in our case 661794 samples per song)

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



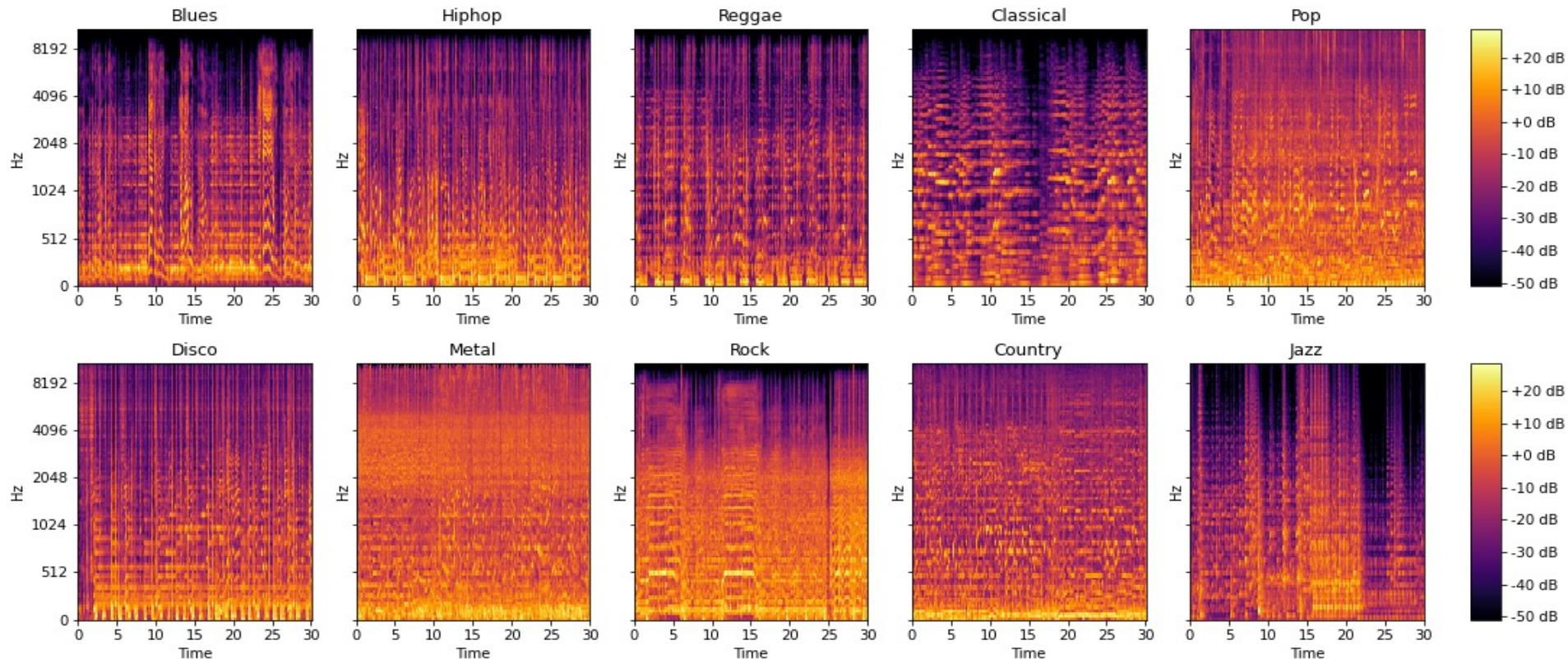
# Feature extraction: Chroma features

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



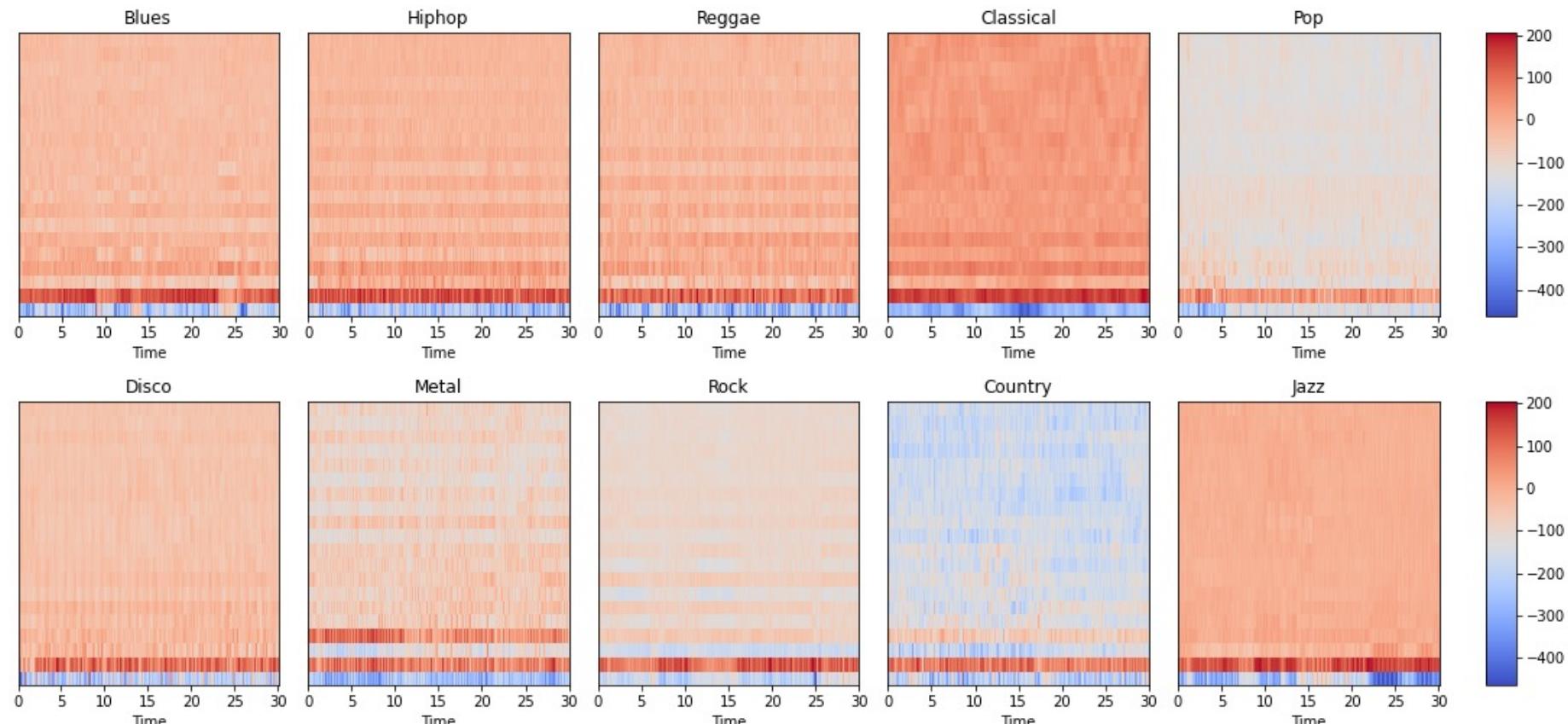
# Feature extraction: Mel spectrograms

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



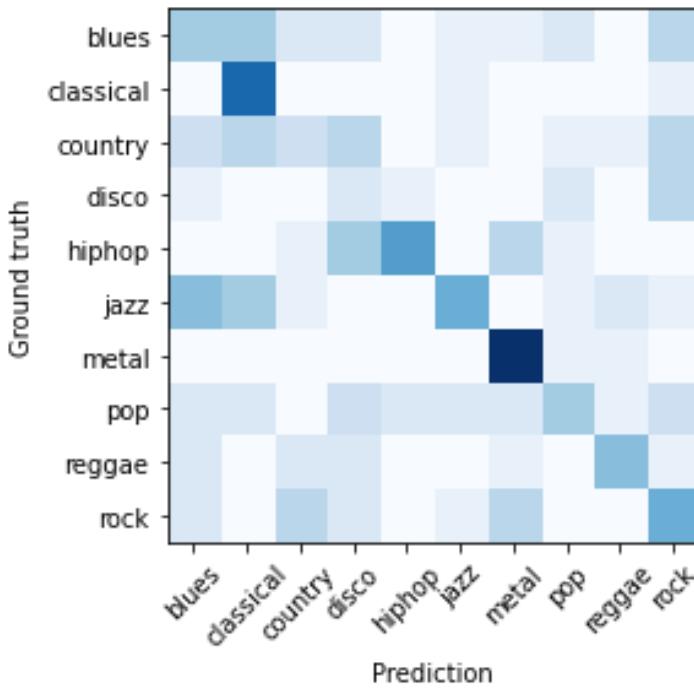
# Feature extraction: MFCC spectrograms

- 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.



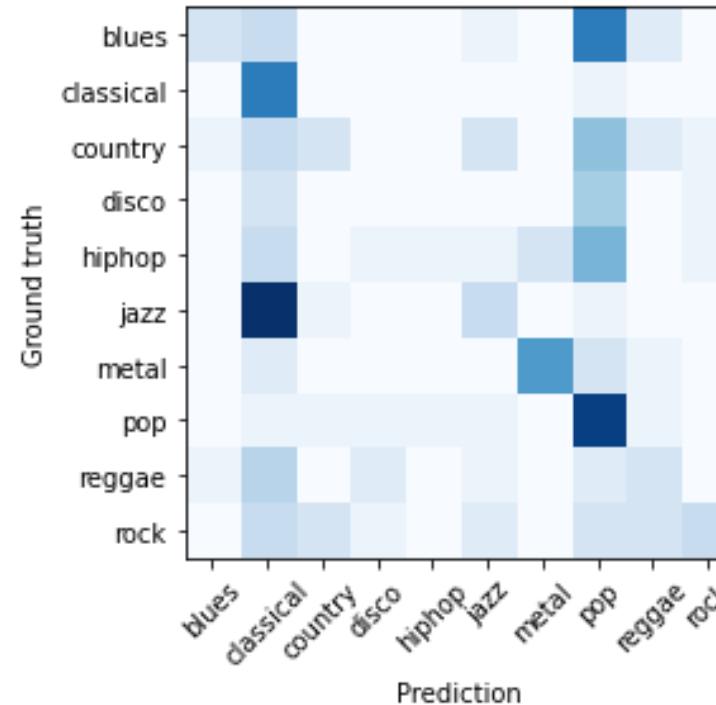
# SVM - Results

Chroma features



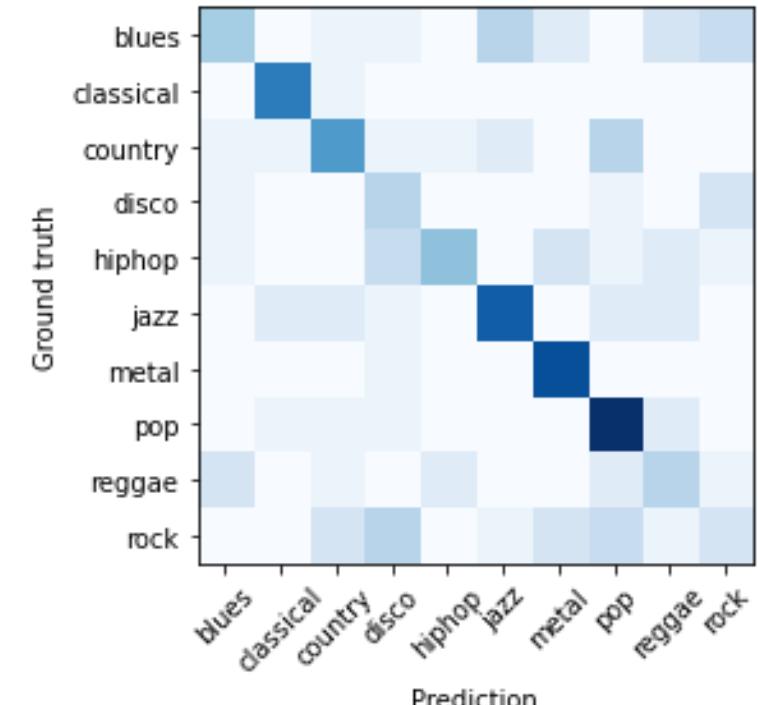
Accuracy: 38%

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

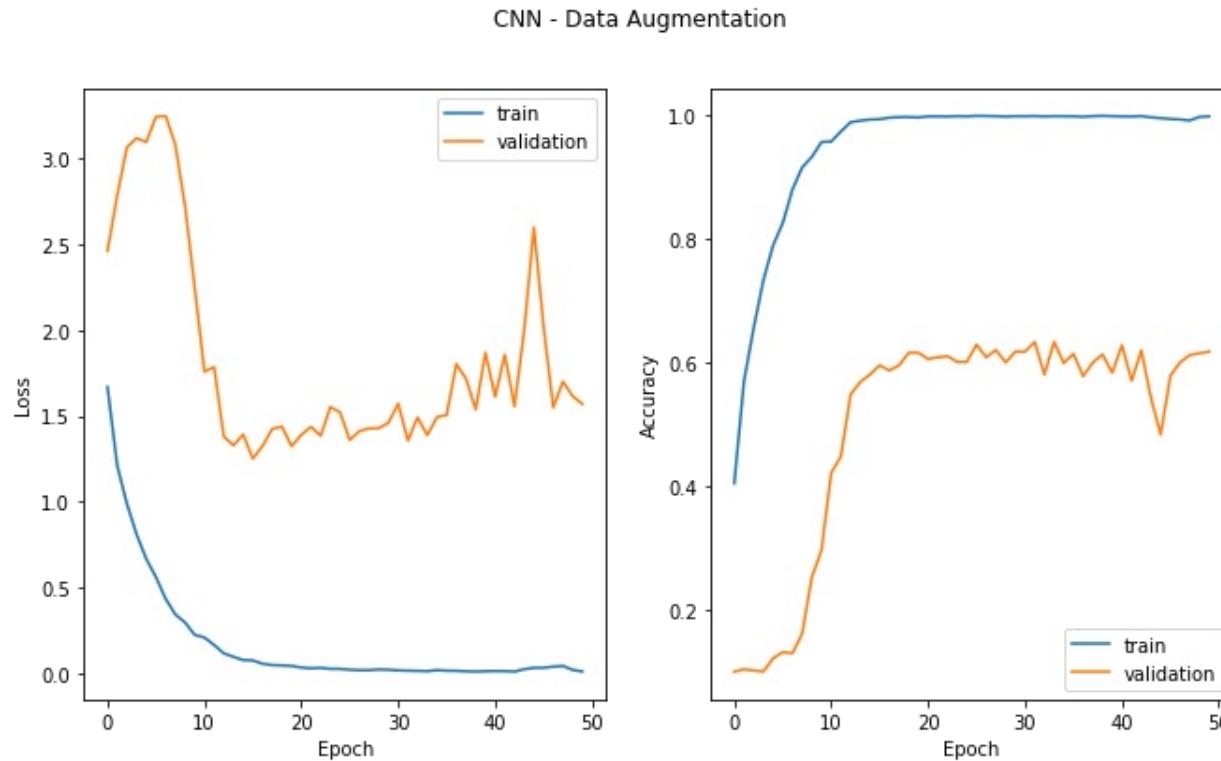
## Audio signals - augmentation

- 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

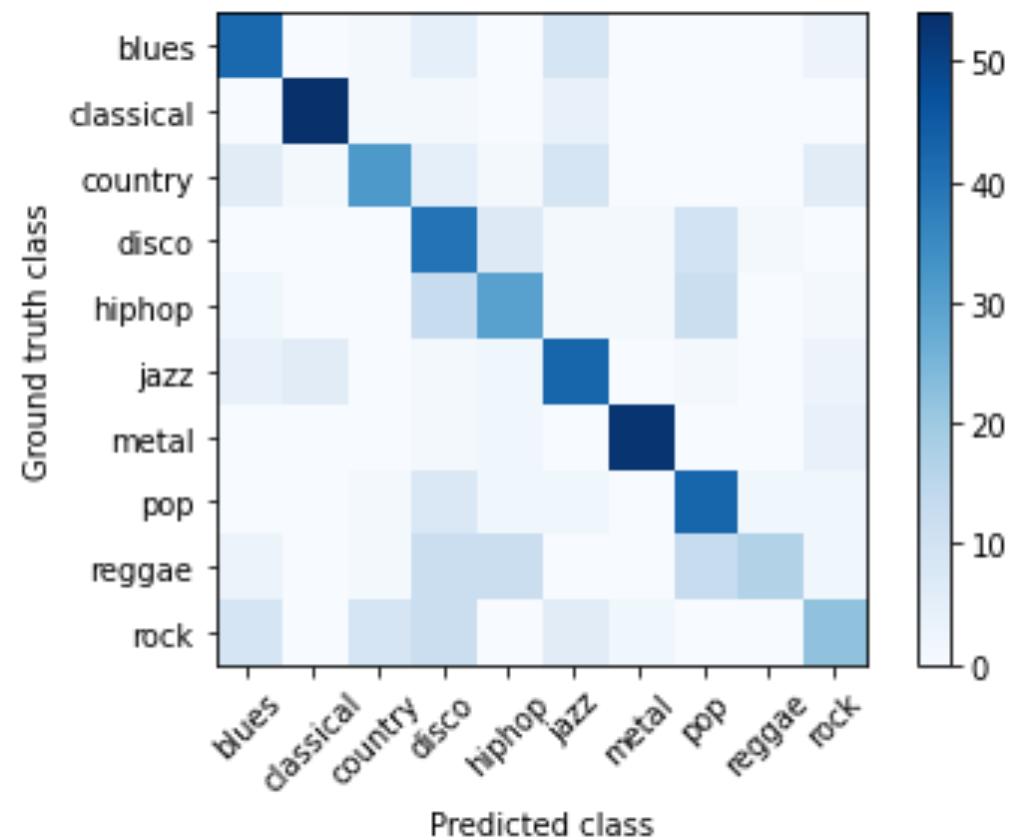
- Input layer with shape (224,224,3)
- 5 blocks of:
  - Convolutional, Batch Normalization and MaxPooling layers, which extract features from the input image
  - ReLU activation
  - Fully connected (dense) layer with **softmax activation**, which outputs the class probabilities
  - Increasing number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

# Test set - evaluation

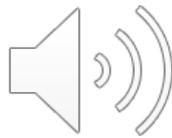
- ❑ Accuracy = 63%
  - ❑ Top 3 Accuracy = 88%

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

Top 1 accuracy: 50%

Top 2 accuracy: 100%

Probability	Genre
0.382395	blues
0.318152	hiphop
0.104593	country

segment 4

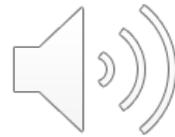
Probability	Genre
0.930740	disco
0.053135	blues
0.008810	country

segment 5

Probability	Genre
0.336893	country
0.200564	blues
0.178876	rock

segment 6

# Demo



*Song out of the test set – downloaded from internet*

```
Greta Van Fleet - Highway Tune
Expected genre: Rock

predict_genre('/content/drive/MyDrive/Second Segment')
1/1 [=====] - 0s 2
Index  Probability  Genre
0      0.681653    blues
1      0.257699    rock
2      0.056348    disco
```

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

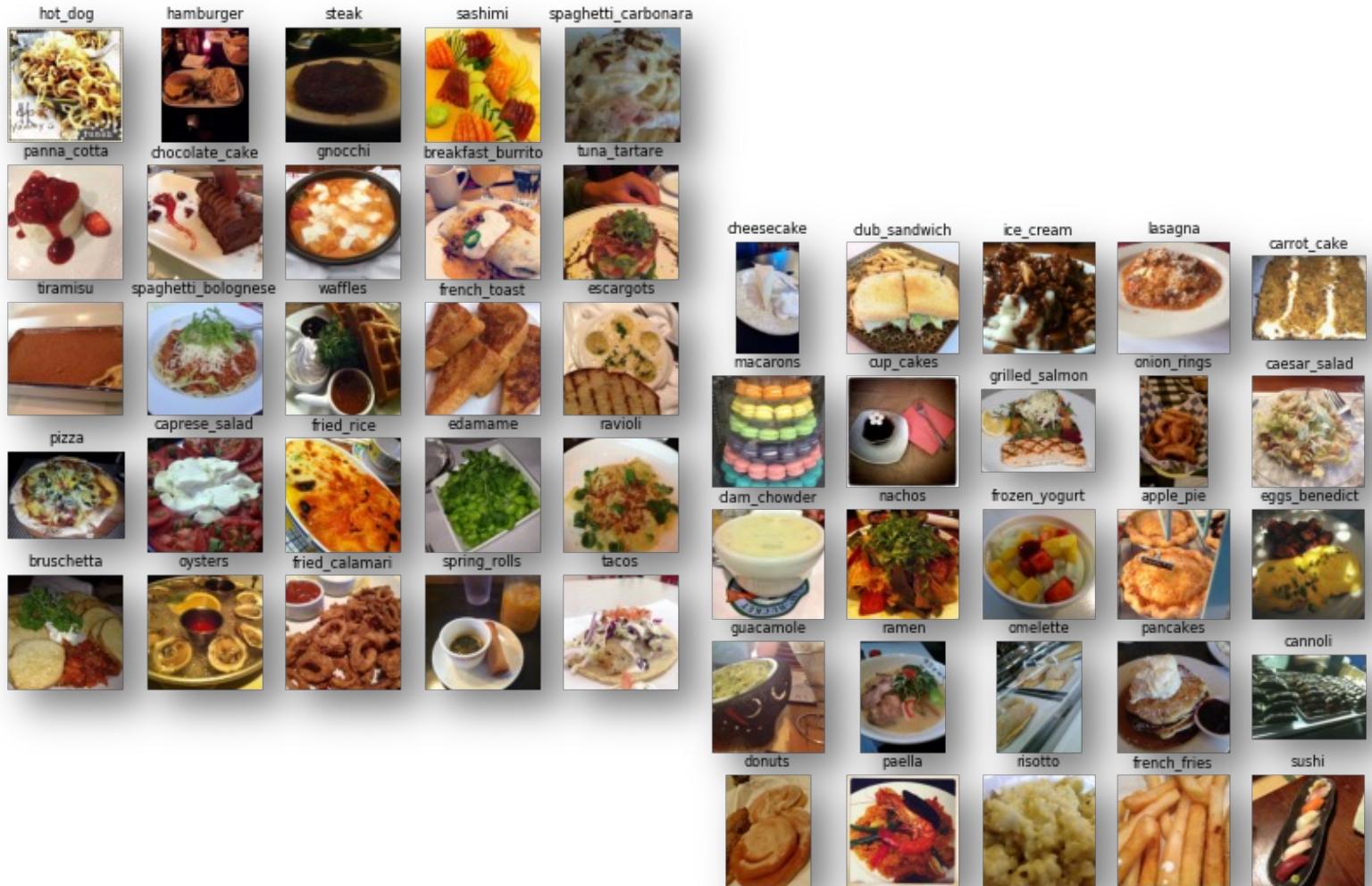
Random sample to 50 classes

- 70% training set
- 30% test set

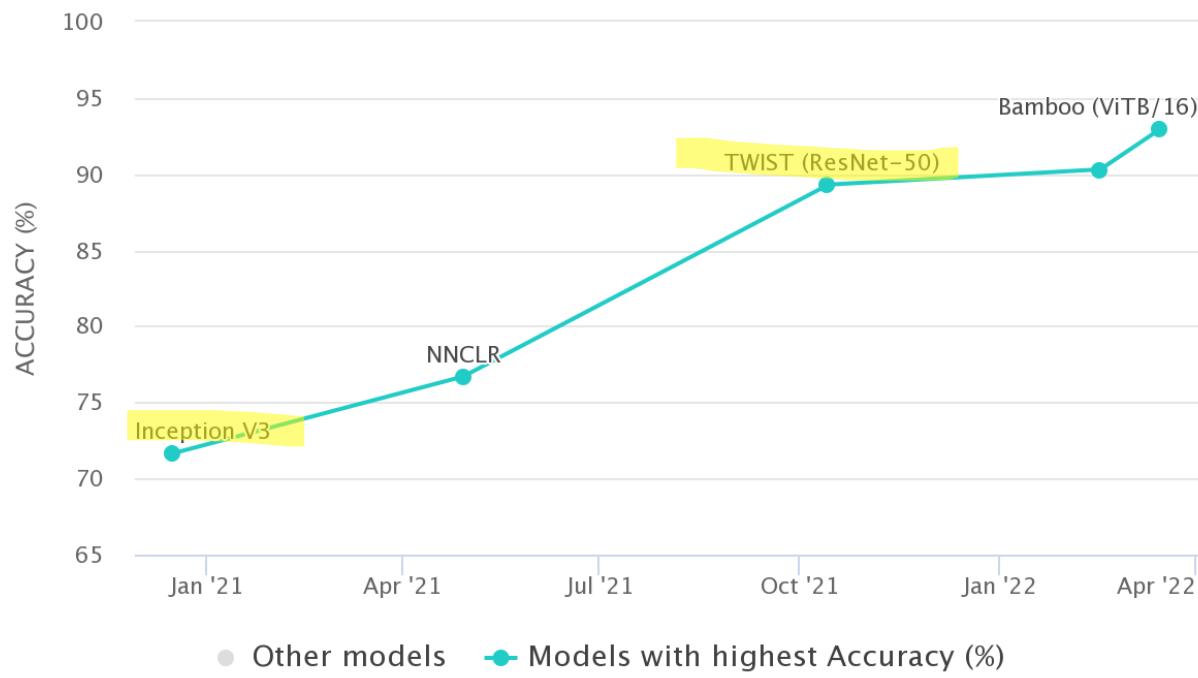
Training set splitted into

- 80% train
- 20% validation

- Different framing
- Different brightness
- Presence of wrong images



# Transfer Learning



ResNet50  
vs  
InceptionV3

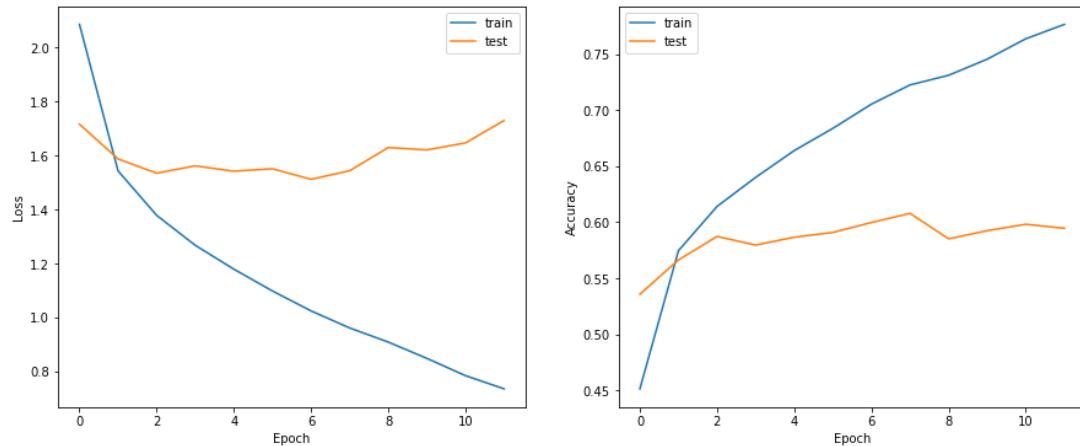


# Transfer Learning – InceptionV3

## 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



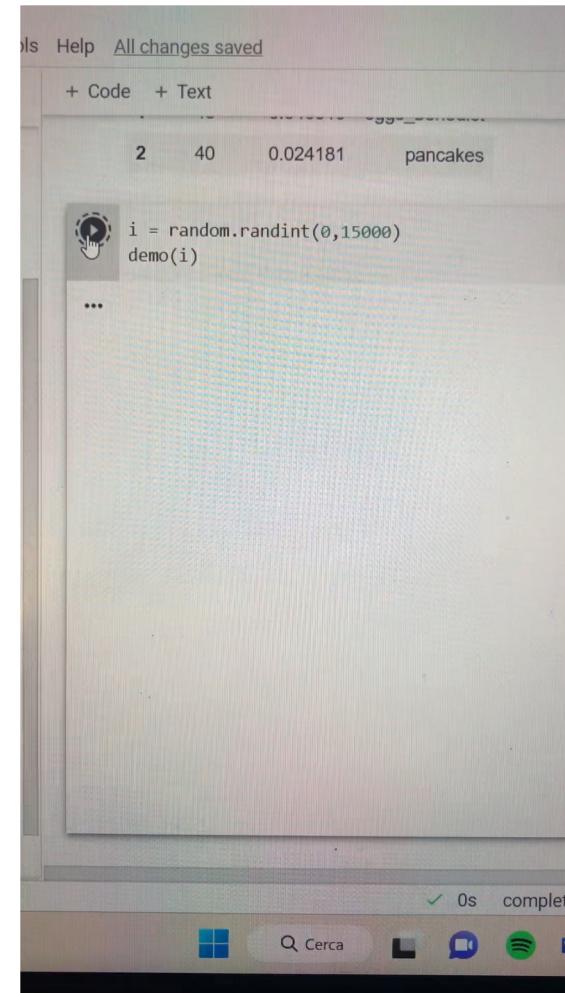
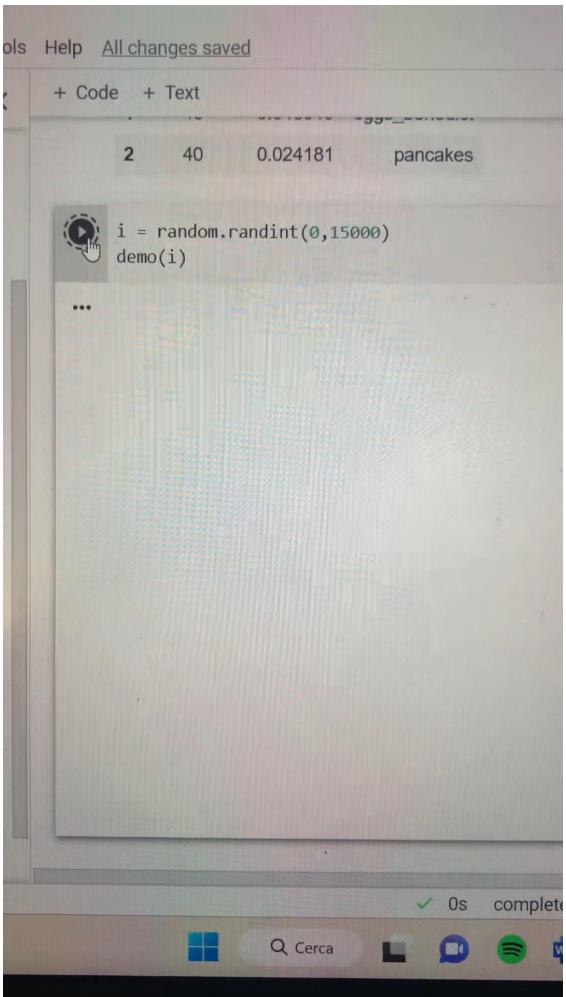
## Final Architecture

- ❑ 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

Confusion Matrix																																																					
apple_pie	breakfast_burrito	caesar_salad	cheesecake	carrot_cake	clam Chowder	club_sandwich	cup_cakes	donuts	edamame	eggs_benedict	escargots	french_fries	french_toast	fried calamari	fried_rice	frozen_yogurt	gnocchi	ice_cream	lasagna	macarons	nachos	omelette	onion_rings	oysters	paella	pancakes	panna_cotta	pizza	ramen	ravioli	risotto	sashimi	spaghetti_bolognese	spaghetti_carbonara	spring_rolls	steak	sushi	tacos	tiramisu	tuna_tartare	waffles												
apple_pie	112	5	0	0	4	0	18	5	2	0	2	1	5	0	4	29	3	1	1	4	2	3	0	2	11	1	0	0	0	7	6	2	1	22	0	3	0	2	0	2	1	1	1	0	4	0	6	1	1	39	1	2	2
breakfast_burrito	2	132	0	0	3	1	3	2	0	11	1	0	1	3	0	2	11	1	0	0	0	7	6	2	1	2	0	2	1	1	1	0	4	0	6	1	1	39	1	2	2												
trussetta	0	1	116	3	2	6	4	3	1	1	5	1	2	1	4	6	0	13	0	0	6	8	0	1	3	2	5	0	9	2	1	1	0	1	0	15	0	16	0	1	2	0	3	2	10	21	1	14	4				
caesar_salad	0	2	2	156	1	0	1	1	0	19	0	0	1	2	0	2	2	5	4	3	4	12	1	1	0	0	10	2	0	1	2	0	2	0	3	6	11	2	2	3	2	1	3	21	1	6	1						
cannoli	4	4	2	0	193	1	3	4	3	3	1	3	10	0	1	0	0	13	0	0	1	1	4	0	1	1	4	0	2	0	3	1	2	1	2	1	0	4	2	0	0	1	4	8	1	4	3	4					
caprese_salad	0	0	19	8	8	92	0	6	3	2	10	3	3	0	10	6	0	8	1	0	4	1	10	1	4	0	1	1	3	0	3	0	2	16	6	2	9	2	6	0	0	2	0	14	5	1	14	10					
carrot_cake	4	1	3	0	7	0	167	20	4	0	3	9	0	0	0	0	10	0	1	1	2	0	0	2	2	4	0	5	2	1	1	1	6	1	1	0	2	1	0	0	1	2	4	0	23	4	5						
cheesecake	4	1	2	0	4	3	9	137	12	1	3	7	2	1	0	2	0	12	0	0	5	0	2	0	1	0	9	2	4	1	0	0	0	3	28	2	0	0	1	0	1	5	2	0	20	1	12						
chocolate_cake	3	0	0	0	12	1	18	17	156	2	0	9	2	0	0	1	0	8	0	0	3	0	2	0	1	0	8	0	3	0	0	1	3	0	1	9	1	0	1	2	0	0	1	20	2	2							
clam_chowder	3	1	0	0	0	0	0	1	3	250	0	1	1	0	2	3	2	0	0	2	0	0	0	0	6	0	1	0	2	0	2	0	2	4	6	4	0	0	0	0	0	0	1	0	0	1	0	0					
club_sandwich	2	1	2	3	0	0	1	0	0	222	0	0	0	0	11	9	0	0	1	0	2	1	11	2	3	1	0	0	5	1	2	0	1	0	0	0	0	0	0	1	2	1	7	0	1	5							
cup_cakes	1	0	1	0	6	0	7	4	4	2	0	217	8	0	2	0	0	0	8	0	0	1	0	9	0	8	1	1	0	0	0	1	3	0	7	0	3																
donuts	5	0	4	0	11	1	2	3	1	18	171	0	1	1	5	11	0	0	0	1	1	10	0	15	2	1	16	1	0	3	2	1	0	0	0	0	2	5	0	1	0	3											
edamame	0	0	1	0	1	1	0	1	0	3	0	0	273	0	1	1	0	0	1	2	1	0	0	2	0	1	0	0	0	0	1	0	1	0	0	5	0	0	1	0	2	0											
eggs_benedict	2	0	8	1	0	3	0	3	2	2	1	0	207	1	0	12	0	0	2	1	9	0	1	2	3	2	2	3	2	1	1	0	6	0	1	3	0	0	0	3	3	4	0	1	2								
escargots	2	0	2	1	0	1	2	0	0	4	1	2	8	0	3	186	0	6	0	1	7	1	0	2	0	0	1	8	0	1	1	0	3	2	1	1	5	6	14	4	1	1	0	1	6	1	5	3	3	3			
french_fries	1	0	0	0	0	0	0	0	15	0	2	0	0	1	252	4	3	0	3	0	1	1	0	1	0	0	1	2	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0			
french_toast	6	1	2	0	3	0	4	2	4	0	1	4	0	1	207	0	0	1	5	0	0	3	4	0	1	9	3	1	0	4	1	1	1	1	0	1	3	5	1	3	1	8											
fried_calamari	0	1	5	0	0	0	2	1	1	2	0	1	0	2	6	5	163	3	1	6	12	2	1	0	1	1	0	8	2	35	1	3	1	1	0	2	5	9	0	0	2	2	2	1	7	0	0	4					
fried_rice	3	3	0	1	1	0	0	0	1	2	0	0	0	0	0	3	177	1	0	2	1	1	0	4	0	3	3	1	0	7	0	0	1	2	2	1	0	1	2	2	1	7	0	0	4	1	0	0					
frozen_yogurt	2	1	1	0	2	0	0	0	1	5	0	4	2	0	0	1	1	239	1	0	4	0	0	23	0	0	0	2	0	1	1	0	0	2	1	0	0	0	2	0	0	1	0	2									
gnocchi	2	0	10	1	2	0	0	2	8	1	0	1	6	7	0	8	4	3	0	124	7	1	0	1	0	6	0	3	1	0	1	2	0	2	4	5	49	21	0	0	2	0	4	1	2	0	7	1					
grilled_salmon	5	0	3	3	2	3	2	1	1	10	0	1	0	2	4	0	22	2	0	2	0	3	136	1	2	3	0	6	0	0	5	0	3	1	0	2	0	6	4	3	2	0	5	26	4	5	1	13	5				
guacamole	0	1	1	4	0	0	0	0	1	3	1	0	1	0	1	1	2	0	288	1	0	1	1	8	2	1	0	3	0	1	0	1	1	4	0	0	0	3	0	13	0	6	1										
hamburger	0	0	2	0	2	1	1	0	0	5	6	10	2	0	1	2	0	206	5	2	0	1	2	1	6	1	0	5	0	1	0	0	0	0	0	2	6	0	6	1	4	1											
hot_dog	2	3	1	0	4	1	2	1	11	1	0	0	0	7	2	1	1	0	0	1	3	13	208	1	0	1	3	1	5	0	0	0	1	0	1	0	1	0	8	0	1	12	0	0	3								
ice_cream	4	1	1	0	5	0	2	1	3	4	0	5	2	0	1	1	0	27	0	0	2	2	1	203	1	0	2	0	5	2	1	2	2	1	0	1	0	0	0	1	1	3	4	1	6								
lasagna	6	3	3	0	4	2	2	1	5	6	1	0	0	1	1	11	0	0	2	1	5	2	0	2	1	1	145	0	3	19	0	0	0	0	24	3	16	9	0	4	0	1	6	0	4	2	2	4	2				
macarons	0	0	0	1	0	2	0	2	1	7	0	0	0	1	0	0	257	0	1	0	0	0	1	2	0	0	1	0	0	1	2	0	0	0	0	3	0	2	1	2	1												
nachos	1	1	2	7	3	1	0	0	1	4	0	8	1	1	5	0	20	1	2	4	1	149	6	3	1	4	0	1	2	8	7	0	1	0	2	3	0	28	1	0	7	1											
omelette	3	10	5	3	2	1	0	3	1	31	0	3	3	0	16	2	13	0	5	99	3	0	1	3	0	14	1	6	4	0	3	1	5	8	3	8	1	4	0	0	1	2	0	0	1	0	2	0					
onion_rings	3	1	0	0	0	0	0	0	2	3	0	2	0	0	5	6	9	0	1	0	0	2	5	0	0	247	0	0	1	0	0	1	2	0	0	2	0	0	1	0	2	0	0	1	0	2	0						
oysters	0	0	1	0	5	0	0	1	2	1	0	0	0	1	0	0	225	3	0	2	2	1	2	1	2	0	0	0	2	6	6	0	1	0	0	2	2	6	0	1	0	1	0	1	0	0	1	0	0	1			
oysters	0	1	2	1	6	2	3	0	0	1	46	0	0	1	2	1	4	0	254	5	4	0	3	1	0	1	0	1	0	196	0	0	10																				

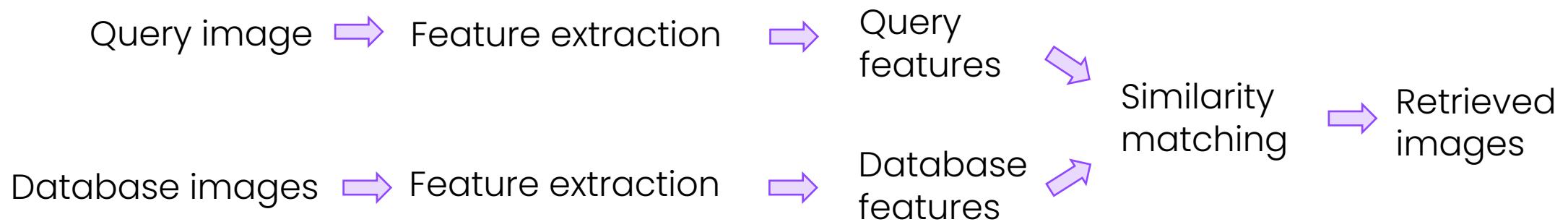
# Demo



# 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  $c$  ( $=100$ )

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

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

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

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:**  $\text{Min } P_c(1) = 22\%$



**Edamame:**  $\text{Max } P_c(1) = 95\%$

# Evaluation : TEST 1

Query image



french\_fries

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

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

10 most similar images retrieved

1: french\_fries



2: french\_fries



3: french\_fries



4: french\_fries



5: french\_fries



6: french\_fries



7: french\_fries



8: french\_fries



9: french\_fries



10: french\_fries



# Evaluation : TEST 2

Query image



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

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

10 most similar images retrieved



# Evaluation : TEST 3

Query image – *out of the test set*

frozen\_yogurt



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

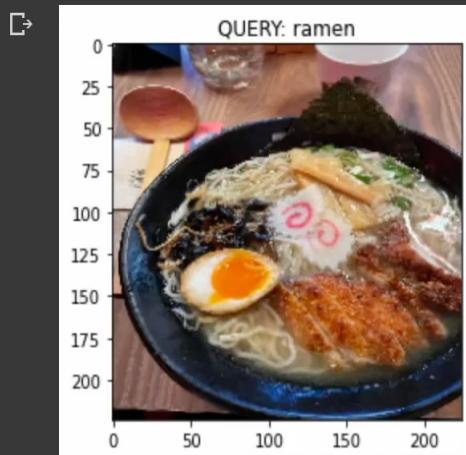
$$\overline{P_c(1)} = 58\%$$

## 10 most similar images retrieved



# Demo Image Retrieval

```
[1]: 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)
```

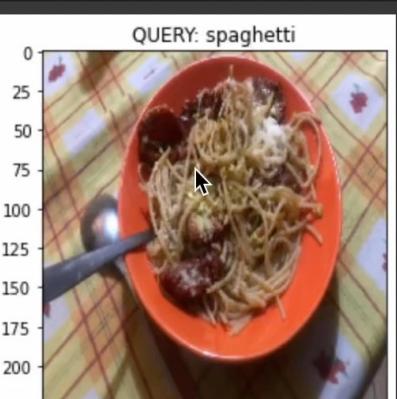


# Demo Image Retrieval

+ Code + Text ✓ RAM Disk

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

QUERY: spaghetti



```
[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))

1: spaghetti_carbonara	2: spaghetti_bolognese	3: spaghetti_bolognese	4: spaghetti_bolognese	5: spaghetti_carbonara	6: caesar_salad	7: risotto	8: spaghetti_bolognese	9: ramen	10: spaghetti_carbonara
------------------------	------------------------	------------------------	------------------------	------------------------	-----------------	------------	------------------------	----------	-------------------------

# 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-b4052befc7e2>
- G.Tzanetakis and P. Cook, 2002, GTZAN dataset, <https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification>