COVID-19 Search Engine

Information retrieval project

Marco Ghezzi - 07523A

Matteo Limoncini - 983857

November 2, 2022

Aim of the project

- Creating a system able to find images that are similar to a given query: it measures the similarity of the two images based on the similarity of the properties of their visual components.

- Considering different approaches for building a content-based medical image retrieval system

- Comparing their results based on classification metrics and computational time

Research question and methodology

Two different Machine Learning approaches:

- supervised learning approach: binary classification
- unsupervised learning approach: clustering

    Two techniques for feature extraction:

    - Principal component analysis (PCA)
    - Convolutional neural network (CNN)

    One partition-based clustering algorithm: *k*-means

Dataset

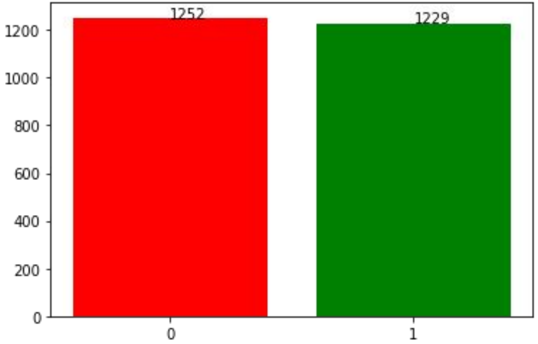
2482 computed tomography scans, collected from real patients in hospital of Sao Paulo, Brazil.

The dataset is balanced, half of them tested positive for COVID-19.

Data preprocessing:

- resize all images:
  required because there are a lot of images with different sizes

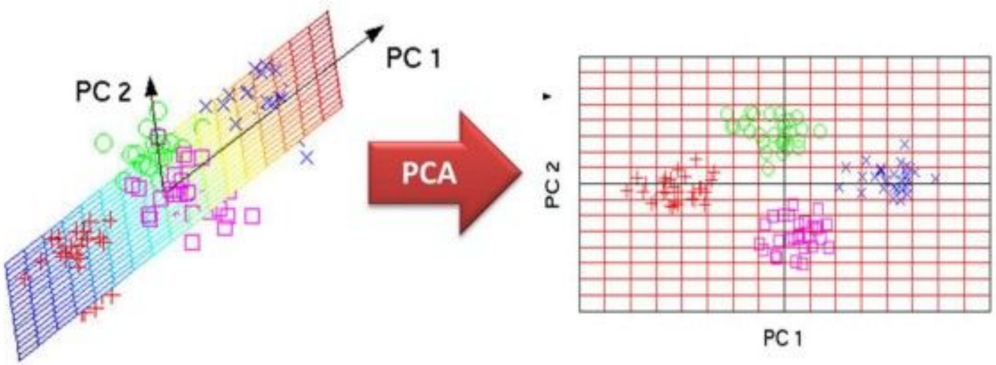
- normalize pixel values between 0 and 1
  needed to reduce the computation complexity



Feature extraction: Principal Component Analysis

PCA tries to reduce the number of variables of a data set, while preserving as much information as possible.

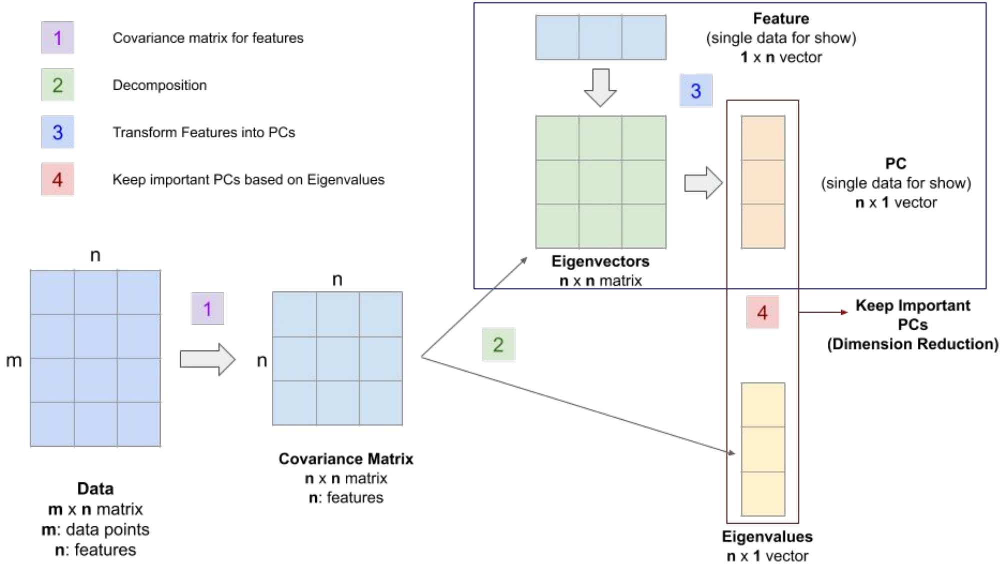
PCA identifies important relationships in the data; it transforms existing data based on these relationships and quantifies the importance of these relationships.



Feature extraction: Principal Component Analysis

PCA can be divided into four steps:

1) We identify the relationship among features through a Covariance Matrix.
2) Through the linear transformation or eigendecomposition of the Covariance Matrix, we get eigenvectors and eigenvalues.
3) Then we transform our data using Eigenvectors into principal components.
4) Lastly, we quantify the importance of these relationships using Eigenvalues and keep the important principal components.



1 Covariance matrix for features

2 Decomposition

3 Transform Features into PCs

4 Keep important PCs based on Eigenvalues

Data
m x **n** matrix
m: data points
n: features

Covariance Matrix
n x **n** matrix
n: features

Feature
(single data for show)
1 x **n** vector

Eigenvectors
n x **n** matrix

PC
(single data for show)
n x **1** vector

Keep Important PCs (Dimension Reduction)

Eigenvalues
n x **1** vector

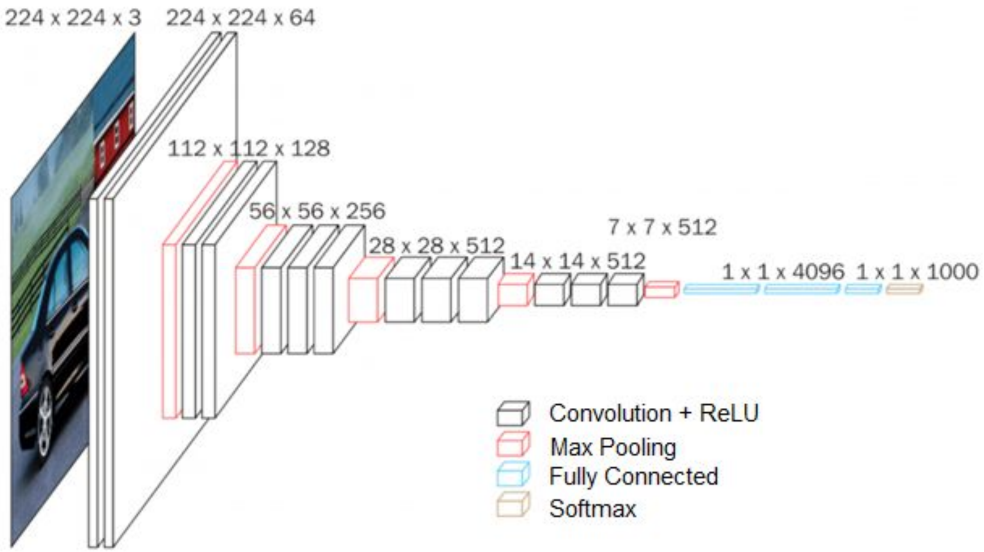
Feature extraction: Convolutional Neural Network

Use a pretrained Convolutional Neural Network as feature extraction method.

VGG16, a CNN model used for image classification and object localization.

The last layer of the VGG16 is removed in order to perform feature extraction

The new last layer is the fully connected layer, used as a feature vector



Partition based algorithm: *k*-means

k-means is a simple, relatively fast and easy scalable partition based algorithm.

Its objective is to minimize the average squared Euclidean distance of images from their cluster centers.

k-means needs to know in advance the number of clusters, it's sensitive to outliers and can consider only linear boundaries, so it's not optimal for clusters with non linear structure.

Another problem is that *k*-means has a random initialization so can generate different clusters in different run.

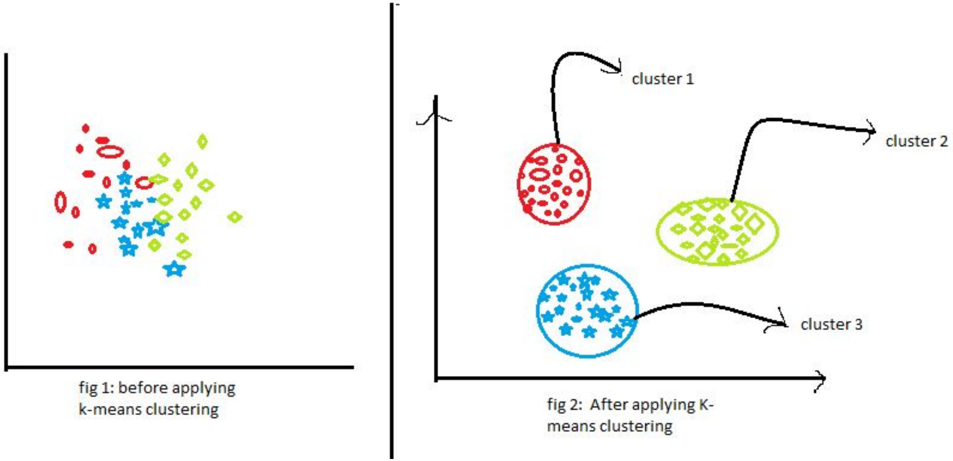
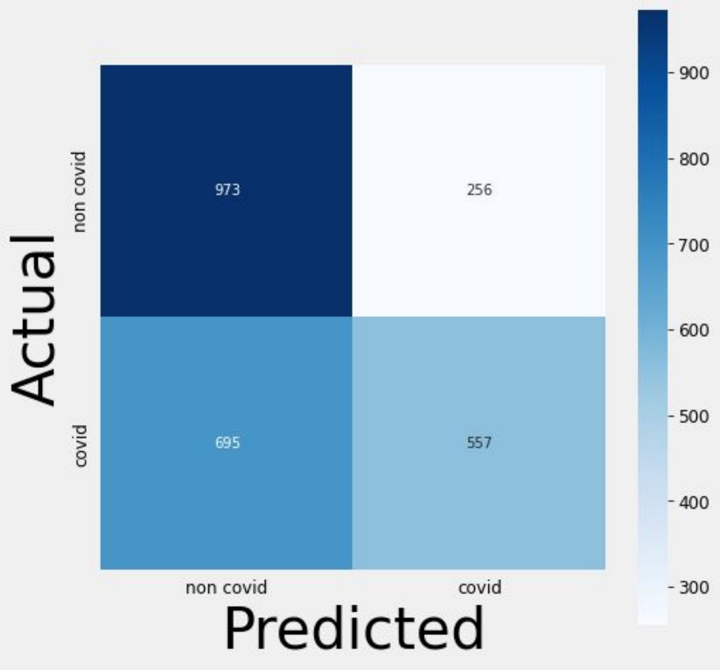


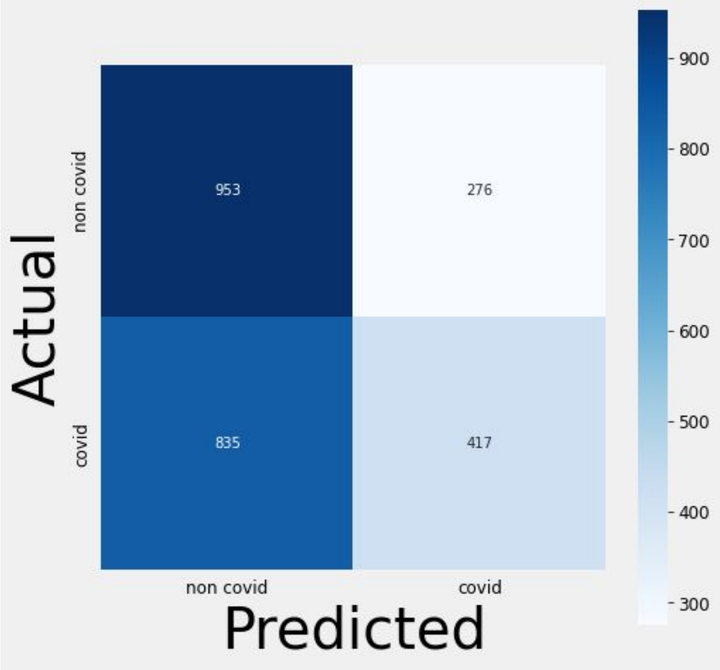
fig 1: before applying k-means clustering

fig 2: After applying K-means clustering

Experimental results: confusion matrix clustering

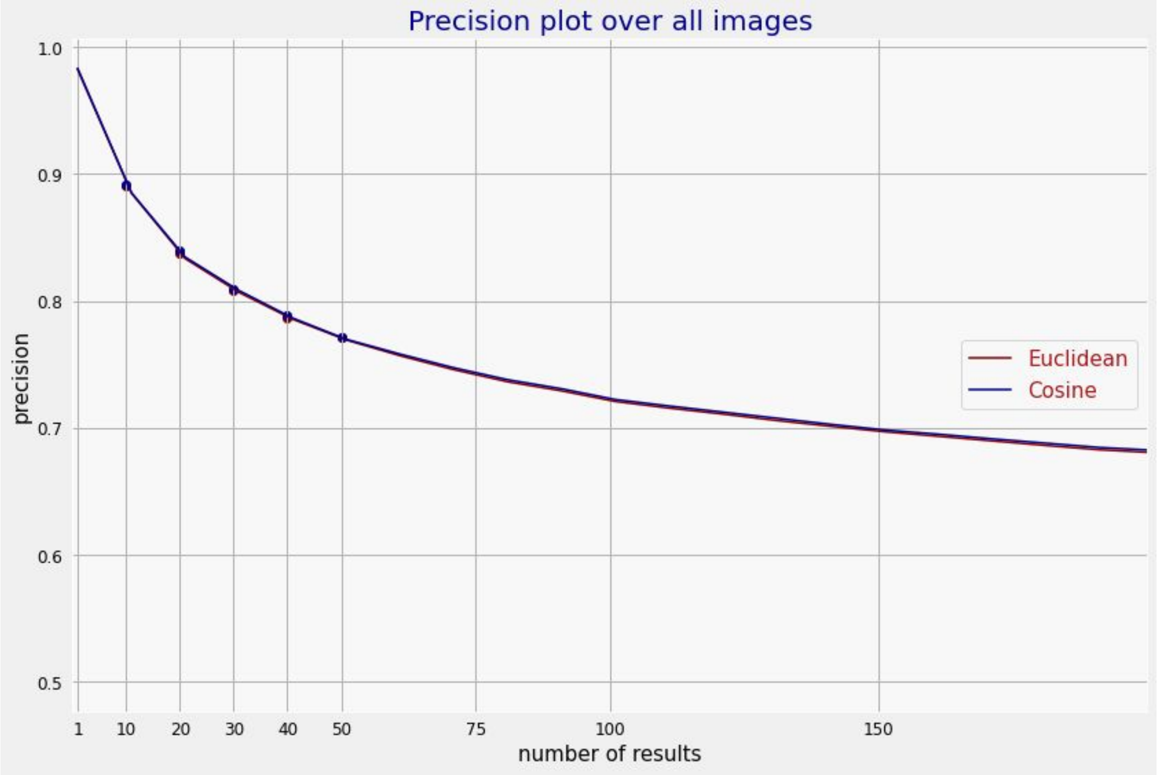


Confusion matrix clustering with PCA

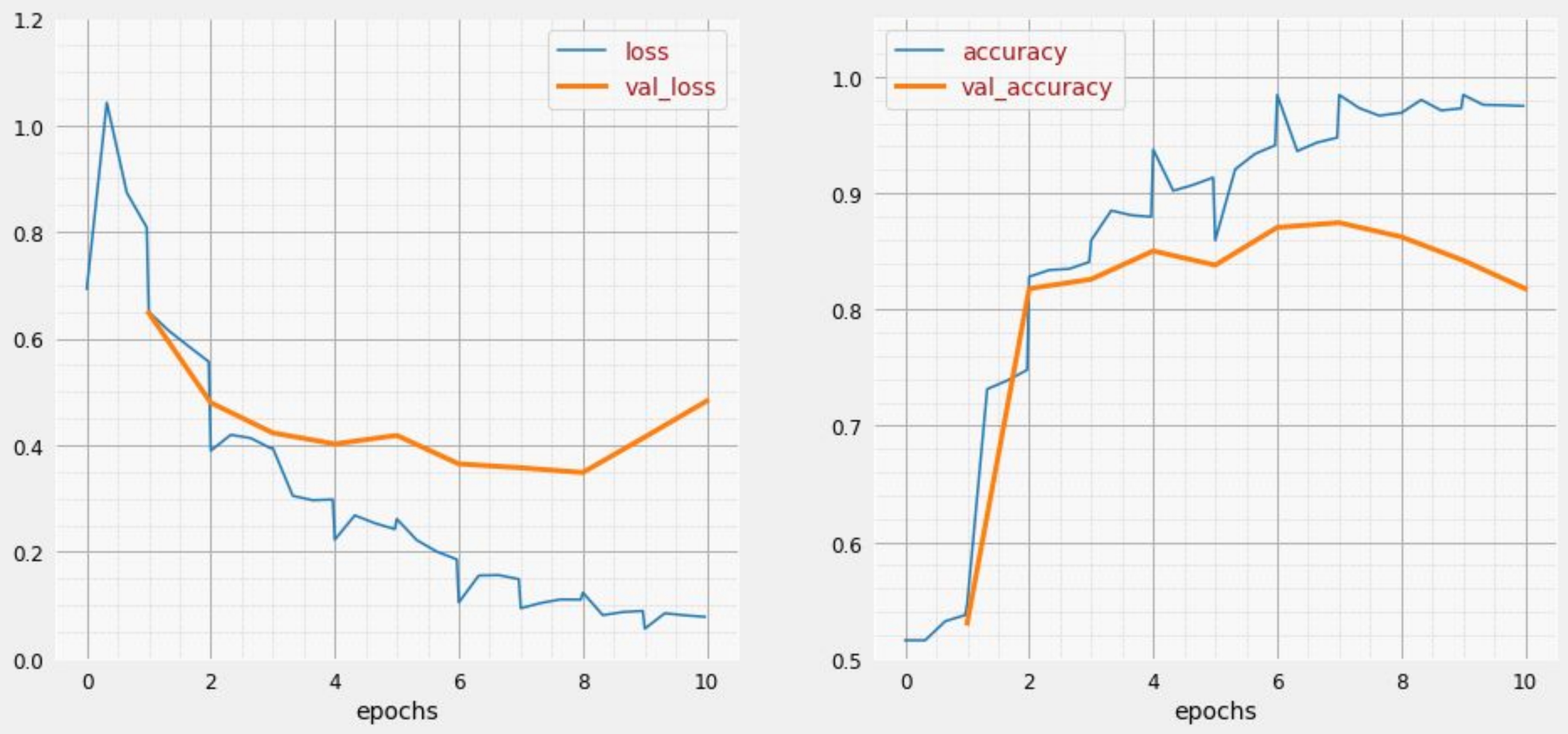


Confusion matrix clustering with CNN

Experimental results: precision curve similarity



Experimental results: binary classification with CNN



Experimental results: computational time

Classification with CNN	≈ 53 sec
Clustering with PCA and *k*-means	≈ 3 min
Clustering with VGG and *k*-means	≈ 2 min
Image similarity- Euclidean	≈ 2 min
Image similarity - cosine	≈ 6 min

Conclusion and future work

Supervised learning with CNN produces better results using less training time than unsupervised learning.

Unsupervised learning is useful if there are images unlabeled.

Possible improvements:

- Improve data preprocessing
- Introduce data augmentation
- Improve explainability
- Introduce new feature extraction methods
- Introduce hierarchical clustering algorithms or density based clustering algorithm