Image and Speech Recognition

Fruit Quality Evaluation

2019/2020

Team members:
Burak Can Buyukbas
Tetiana Bakai
Ivana Škorupová

Contents

Task description

The purpose of this project is to classify three different types of fruits (apples, bananas and oranges) into two categories: fresh and rotten.

We have chosen this topic because we think that it is a very interesting field and it can be used for agriculture. The effectiveness of quality control increases as the number of people observing the product and process increases. Clearly there are advantages involving everyone who is in contact with the product and process in some aspect of quality control. The fruit and vegetable processing industries are interested to ensure that the manufacturing process is operating satisfactorily and that the final product complies with its specification.

Also it is a good experience to deal with data provided by kaggle. We have selected Kaggle as our platform to look for the dataset since it is one of the best for different competitions.

The project can be divided into five parts:

1. Research (collective)
2. Preparation of datasets (Tanya, Ivana)
3. Development (collective)
    a. Data Preprocessing (Burak)
    b. Feature Extraction (Tanya, Ivana)
    c. Classification (Burak, Ivana, Tanya)
4. Testing (Burak)
5. Documentation (collective)

We have decided to use Python as it has many great libraries for image processing. There are so many libraries to choose from. OpenCV, Keras, PIL/Pillow, SciPy, scikit-image are few examples of the many that provide developing of image processing.

Algorithms description

Data preprocessing

- **Image segmentation**
  We can divide or partition the image into various parts called segments. It's not a great idea to process the entire image at the same time as there will be regions in the image which do not contain any information. By dividing the image into segments, we can make use of the important segments for processing the image. That, in a nutshell, is how image segmentation works. An image is a collection or set of different pixels. We group together the pixels that have similar attributes using image segmentation. [0]
  - **RGB to grayscale** - Used to convert an image with RGB channels into an image with a single grayscale channel. The value of each grayscale pixel is calculated as the weighted sum of the corresponding red, green and blue pixels. For example: **Y = 0.2125 R + 0.7154 G + 0.0721 B** [4]
  - **Gaussian blur -** In image processing, a Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. [6]
    Standard deviation (sigma) - Standard deviation for Gaussian kernel. The standard deviations of the Gaussian filter are given for each axis as a sequence, or as a single number, in which case it is equal for all axes.
    Using Gaussian filter with **standard deviation** for smoothing the picture to reduce the standard deviation of pixel values in the picture in order to reduce image noise. [5]

Features extraction

- **Local Binary Patterns (LBP)**
  - Local Binary Patterns, or LBPs for short, are a texture descriptor.
  - This local representation is constructed by comparing each pixel with its surrounding neighborhood of pixels.
  - The first step in constructing the LBP texture descriptor is to convert the image to grayscale. For each pixel in the grayscale image, we select a neighborhood of size *r* surrounding the center pixel. A LBP value is then calculated for this center pixel and stored in the output 2D array with the same width and height as the input image.
  - For example, let's take a look at the original LBP descriptor which operates on a fixed *3 x 3* neighborhood of pixels just like this:

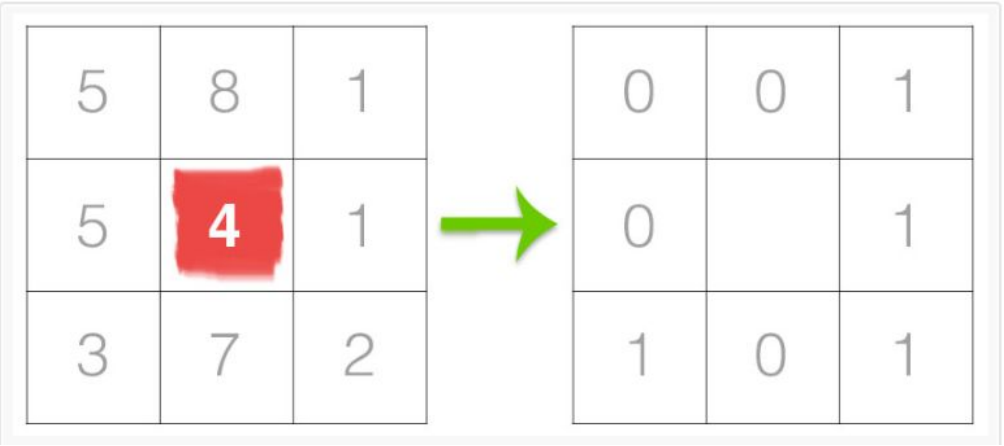


Figure 1: The first step in constructing a LBP is to take the 8 pixel neighborhood surrounding a center pixel and threshold it to construct a set of 8 binary digits.

We take the center pixel (highlighted in red) and threshold it against its neighborhood of 8 pixels. If the intensity of the center pixel is greater-than-or-equal to its neighbor, then we set the value to *1*; otherwise, we set it to *0*. With 8 surrounding pixels, we have a total of *2 ^ 8 = 256* possible combinations of LBP codes.

From there, we need to calculate the LBP value for the center pixel. We can start from any neighboring pixel and work our way clockwise or counter-clockwise, but our ordering must be kept *consistent* for all pixels in our image and all images in our dataset. Given a *3 x 3* neighborhood, we thus have 8 neighbors that we must perform a binary test on. The results of this binary test are stored in an 8-bit array, which we then convert to decimal, like this:

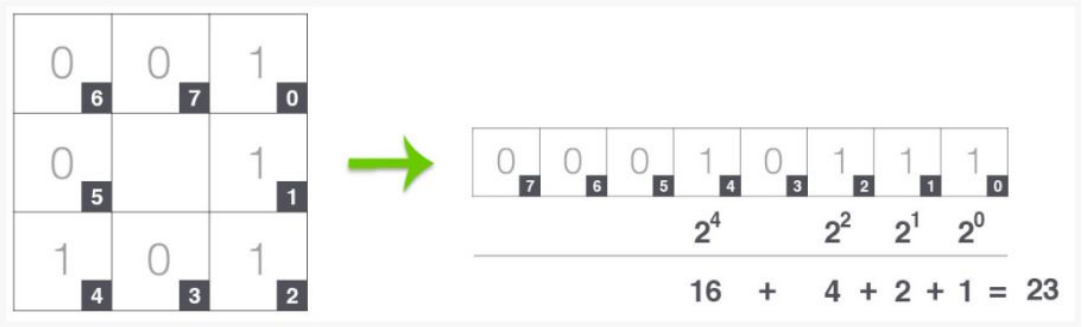


Figure 2: Taking the 8-bit binary neighborhood of the center pixel and converting it into a decimal representation.

In this example we start at the top-right point and work our way *clockwise* accumulating the binary string as we go along. We can then convert this binary string to decimal, yielding a value of 23.

This value is stored in the output LBP 2D array, which we can then visualize below:

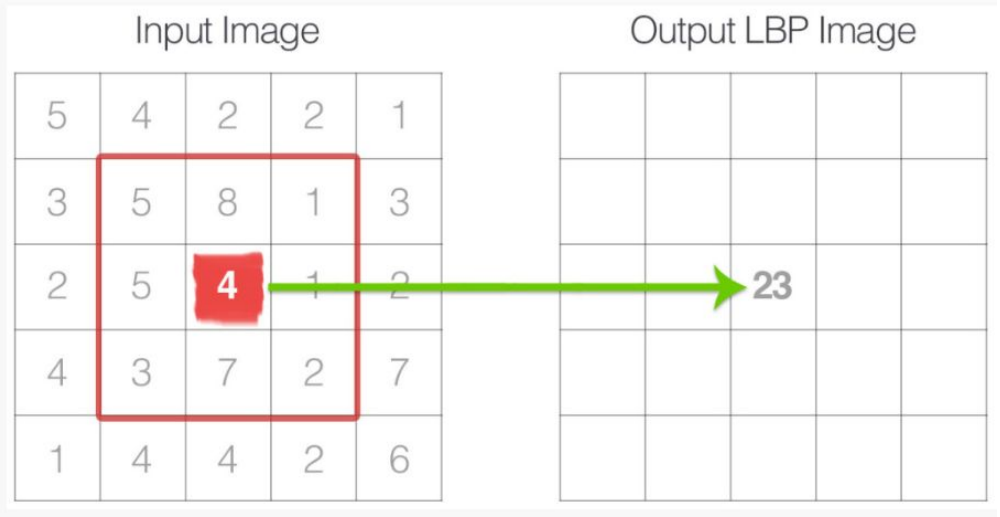


Figure 3: The calculated LBP value is then stored in an output array with the same width and height as the original image.

This process of thresholding, accumulating binary strings, and storing the output decimal value in the LBP array is then repeated for each pixel in the input image. [3]

- **Color Features**

  The factors which influence the customer to reject or choose the fruits and vegetables are color. It is the indirect measurement of quality characteristics like freshness, desirability and variety, maturity and safety which is governed by physical and chemical changes, internal biochemical, microbial occurs in ripeness, growth and postharvest processing and handling stages. The color feature is the first and most widely used visual features in image retrieval and indexing. The color feature has many advantages like high efficiency, ease in extraction of color information from images, size and orientation independent, powerful in representing visual content of images, robust to background complications and powerful in separating images from each other. The RGB color space, HSI space, CIELab space is commonly used for color inspection of quality of fruits and vegetables. Once the color spaces have been specified, color feature can be extracted from images. Various color features have been introduced by many researchers that includes, color correlogram, color coherence vector, color moments, and color histogram, etc. Among them, color moment is simple and effective. The most common moments are mean, standard deviation and skewness. [7]

- **Textural Features**

  Texture measured from group of pixels represents the distribution of elements and surface appearance and is useful in machine vision that predicts surface in form of

roughness, contrast, entropy, orientation, etc. Texture is compatible to maturity and sugar content (internal quality of fruits and vegetables). It is also used to segregate different patterns in images by extracting intensity values between pixels. Texture can be analyzed by quantitative and qualitative analysis. According to quantitative analysis six textural features i.e. contrast, coarseness, line-likeness, directionality, roughness and regularity. According to qualitative analysis four features i.e. contrast, correlation, entropy and energy. Different type of texture features are statistical texture, model based texture, structural texture and transform based texture. Statistical texture, extract matrix (gray level co-occurrence matrix, gray level pixel run length matrix and neighboring gray level dependence matrix) which is based on intensity values of pixels. Model based texture includes fractal model, random field model and autoregressive model. Structure texture includes lines, edges that are constructed by pixels intensity. Transform based texture can be extracted spatial domain images. Because of low computational cost and high accuracy, statistical texture is commonly used. [7]

- **Morphological Features**

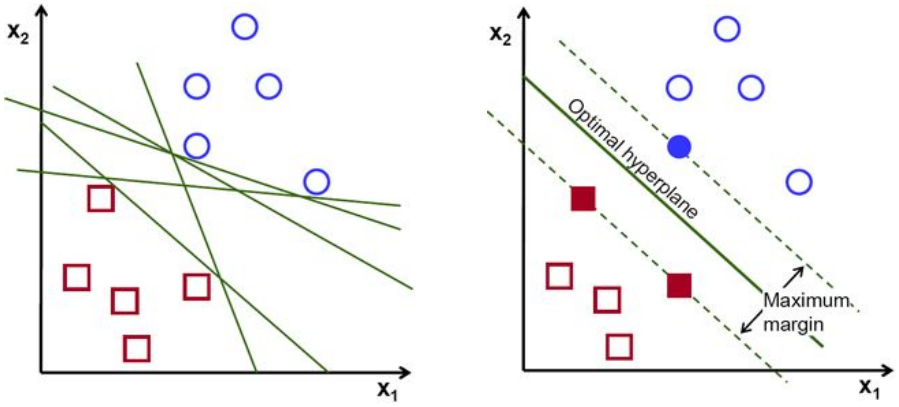
  The morphological features (size and shape) are frequently used for classification of fruits and vegetables. In agriculture industry, the size of fruits and vegetables relates to price, therefore in processing stages different size group are allotted for grading of fruits and vegetables. Spherical and quasi-spherical object size of fruits and vegetables inspection is very easy compare to natural irregularity of complex foods. Quantifying the feature size is measured by using projected area, perimeter, length, width, major and minor axis. These features are broadly used in automatic sorting purpose in industries. The area (scalar quantity) calculates the actual number of pixels in the region. Projected area is acquired by pixels of the area. The distance of two neighboring pixels results in feature extraction. Perimeter (scalar quantity) is the distance between the boundaries of region. No matter what shape or orientation, once the object is segmented area and perimeter are stable and efficient. To quantify the size of fruits and vegetables length and width are used. Since the shape of food products is usually change during processing, the orientation at which length and width are calculated needs to be restored in time. The longest line across the object, obtained by the distance of every two boundary pixels is major axis. The longest line drawn across the object perpendicular to the major axis is minor axis. Shape is a critical visual feature for image content description which cannot be defined precisely because it is difficult to measure the similarity between shapes. The two categories of shape descriptor are: region based (based on integral area of object) and contour based (boundary segmented using local features). Shape features are measured by roundness, aspect ratio and compactness. [7]

Classification

In computer vision system, a wide variety of methods have been developed for classification in food quality evaluation. In the following lines, the chosen algorithms for use in classification part of our project will be described.

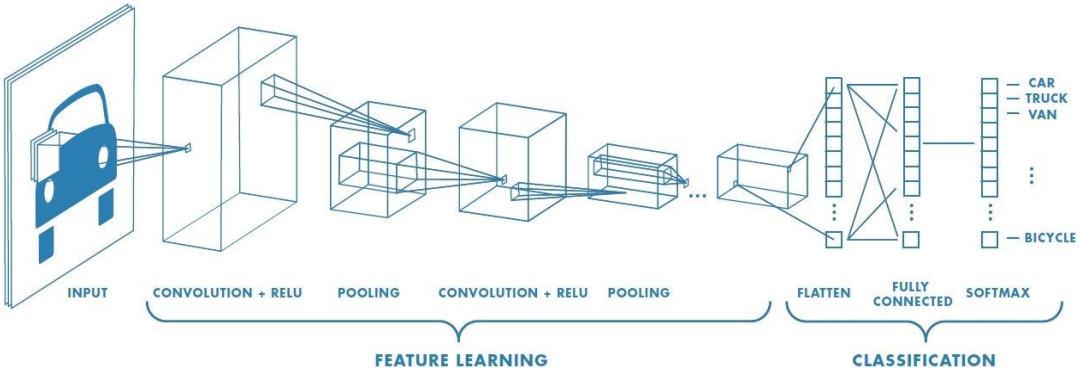
➔ **Support vector machine (SVM)**

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. [1]



➔ **Deep Learning/Convolutional Neural Network (CNN)**

In neural networks, CNNs is one of the main categories to do image classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used. CNN image classifications takes an input image, process it and classify it under certain categories. Computers see an input image as the array of pixels and it depends on the image resolution. Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. [2]



Datasets description

Our dataset contains images of fresh and rotten fruits (apples, bananas and oranges) for the training and testing. The format of our images is .png . We used around 1000 pictures of fresh and rotten fruits of each type for the training and testing.

All our images are on a white background. We have different positions of fruits at the images and also the color of the particular type of fruit is not the same in all cases. Also, some fruits (mostly apples) have drops of water on them and sometimes they are with green leaves and tails. Some examples of fresh apples have equally distributed dots of another color on their skin and they might have a glow on their surface. Also some pictures of bananas contain different amounts of a particular fruit. [8]

Examples of images in the dataset

Fresh apples

Rotten apples

Fresh oranges

Rotten oranges

Fresh bananas

Rotten bananas

Implementation

Model details

```
model = Sequential()
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu', input_shape=(64, 64, 4)))
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Droput(0.25))
model.add(Flatten())
```

```
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(3, activation='softmax'))
opt = adadelta(lr=0.001, decay=1e-6)
model.compile(optimizer=opt,loss=sparse_categorical_crossentropy, metrics=['accuracy'])
reduce_lr=ReduceLROnPlateau(monitor='val_loss',factor=0.2,patience=5, min_lr=0.00001)
model.fit(X_train, Y_train, batch_size=128, epochs=epoch_count, verbose=1,
          validation_data=(X_test, Y_test),  callbacks = [reduce_lr])
```

Present results

Fresh/Rotten Classification Prototype Results:

Fruit Type	**Total Accuracy**	**Fresh Accuracy**	**Rotten Accuracy**
Banana	224/230 (97%)	121/125 (96.8%)	103/105 (98%)
Apple	177/200 (88.5%)	88/102 (86.2%)	89/98 (90.8%)
Orange	189/200 (94.5%)	100/102 (98%)	89/98 (91%)

Models are trained with 300 epochs. Optimizer: Adadelta with 0.001 learning rate

Fruit Type Classification Prototype Results:

SVM 1000 iteration, color and segmentation (without lbp), C=0,0001

Total Accuracy	**Apple Accuracy**	**Banana Accuracy**	**Orange Accuracy**
485/630 (76.98%)	153/204 (75.0%)	195/229 (85.15%)	137/197 (69.54%)

SVM 1000 iteration, color and segmentation (without lbp, removeBackground), C=0,0001

Total Accuracy	**Apple Accuracy**	**Banana Accuracy**	**Orange Accuracy**
515/630 (81.74%)	164/200 (82.0%)	195/236 (82.62%)	156/194 (80.41%)

SVM 1000 iteration, color and segmentation (with lbp), C=0,0001

Total Accuracy	**Apple Accuracy**	**Banana Accuracy**	**Orange Accuracy**
466/630 (73.96%)	135/189 (71.42%)	171/213 (80.28%)	160/228 (70.17%)

SVM 1000 iteration, color and segmentation (without lbp), C=0,001

Total Accuracy	**Apple Accuracy**	**Banana Accuracy**	**Orange Accuracy**
442/630 (70.1%)	91/203 (44.8%)	163/221 (73.7%)	188/206 (91.3%)

2-layer Cnn, 300 epochs, without lbp

Total Accuracy	Apple Accuracy	Banana Accuracy	Orange Accuracy
432/630 (68.5%)	7/185 (3.8%)	219/234 (93.6%)	206/211 (97.6%)

2-layer Cnn 100 epoch, with lbp

Total Accuracy	Apple Accuracy	Banana Accuracy	Orange Accuracy
358/630 (56.8%)	44/207 (21.2%)	151/238 (63.4%)	163/185 (88.1%)

2-layer Cnn 300 epoch, without lbp

Total Accuracy	Apple Accuracy	Banana Accuracy	Orange Accuracy
419/630 (66.5%)	11/196 (5.6%)	221/242 (91.3%)	187/192 (97.3%)

2-layer Cnn, 250 epoch, with lbp

Total Accuracy	Apple Accuracy	Banana Accuracy	Orange Accuracy
402/630 (63.8%)	218/218 (100.0%)	184/225 (81.7%)	0/187 (0.0%)

2-layer Cnn, 300 epochs, without lbp

Total Accuracy	Apple Accuracy	Banana Accuracy	Orange Accuracy
412/630 (65.3%)	6/208 (2.8%)	222/232 (95.6%)	184/190 (96.8%)

Progress

- ~~Convert to console application~~
- ~~Code Cleanup~~
- ~~Improve fruit type classification accuracy(seems like we are overfitting)~~
- ~~Test with different models(more layers etc)~~

Conclusion

To sum up, we got better results with SVM for fruit type classification and CNN for quality classification.

While quality evaluation for each fruit type resulted with good results, fruit type classification needs improvements. Having RGB features(and some cases also LBP) for each pixel causes a large feature matrix and this might be lowering the learning ability of our models. Detecting and using dominant color segments or having better segmentation of areas might increase success in training.

References

[0]-https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/

[1]-https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[2]-https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[3]-https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/

[4]-https://scikit-image.org/docs/dev/auto_examples/color_exposure/plot_rgb_to_gray.html?fbclid=IwAR3ZKtkitwYQra3gumteALPtfa9h1xi89Nyel3wfjbaAVBbQhzCiKGVmlLM

[5]-https://scikit-image.org/docs/dev/api/skimage.filters.html?fbclid=IwAR3EjsuFsDlMwb-Y2cIVnYkY55JyK-9jbGXuneQLbQiAy-dW3Hpunhl7lzU#skimage.filters.gaussian

[6]-https://en.wikipedia.org/wiki/Gaussian_blur

[7]-https://www.sciencedirect.com/science/article/pii/S131915781830209X

[8]-https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification