

# Project 3 - Machine Learning - Hand Gestures

- [Enunciado](#)

## Instructions

The project instructions are located at [ENUNCIADO.pdf](#).

## Installation

Instructions for installing OpenCV on Ubuntu 14.04 or 13.10: [here](#).

Instructions for installing OpenCV on Ubuntu 13.04 or below: [here](#).

Download the [example sets](#) and extract them to the examples folder.

You'll also need Python and Ruby.

## Generating ARFF files

Run `arff_generator.py`, providing the number of frames extracted per example (more than 2 -- more frames provides better but slower classification) and the example sets that will be used.

The following call extracts 6 frames per example, uses the Sets 2, 3 and 5, and saves it all on a file:

```
$ python extractor/arff_generator.py 6 2 3 5 > 6-235.arff
```

## Features

For each example, we extract a number of frames from it.

For example, for 2 frames, we get the first and last frames.

For 3 frames, we get the first, the last and the middle frames.

Then, we extract features for each of these individual frames.

The frame image is segmented based on a HSV range roughly corresponding to skin color range: [0, 30, 60] to [50, 150, 255];

Then, from all the contours in the image (found with `findContours`), we grab the one with the biggest area.

We use this contour to fit a minimum-area rectangle, an ellipse, and a convex hull. We also extract convexity defects between the contour and the convex hull, and the contour's Hu moments.

From these primitives, we can extract all of the features:

```
area
perimeter
convex_hull_area
solidity
rect_center_x
rect_center_y
rect_width
rect_height
rect_angle
rect_aspect_ratio
ellipse_center_x
ellipse_center_y
ellipse_major_axis
ellipse_minor_axis
ellipse_angle
farthest_convex_defect
hu_moment_1
hu_moment_2
hu_moment_3
hu_moment_4
hu_moment_5
hu_moment_6
hu_moment_7
```

These 23 features are extracted for each frame. So, if an example is composed of 10 frames, it'll actually have 230 features.

In the ARFF file, we append a number to each attribute, corresponding to the frame number.

## Results

Please see the presentation.

## Implementing

We implemented a Neural Network algorithm with backpropagation in Ruby.

Take a look at the file `nn/xor.rb` for an example of using it.