Pixel is selected

Appropriate threshold is selected

Consider circle of 16 pixels around the pixel under test

Pixel p is a corner if there are n contiguous pixels which are all brighter than Instensity of p plus threshold, or all darker

High speed test is used to exclude a large number of non corners

Doesn’t reject as many candidates for n<12

Choice of pixels is not optimal because it depends on the ordering of questions and distribution of corner appearances

Results of highspeed test are thrown away

Multiple features are detected adjacent to eachother (tests pass for neighboring pixels)

Machine learning a corner detector

Select image training set

Run fast in every image to find feature points

For every point store pixels around it

Each of the 16 pixel can have one of three states

Compare instensity I of x as x approaches p to I of p

Similar to p (within two sided threshold)

darker , less than threshold

Brighter, greater than threshold

Depending on these states the feature vector P is divided into 3 subsets Pd Ps and Pb

Define boolean Kp which is true if p is a corner

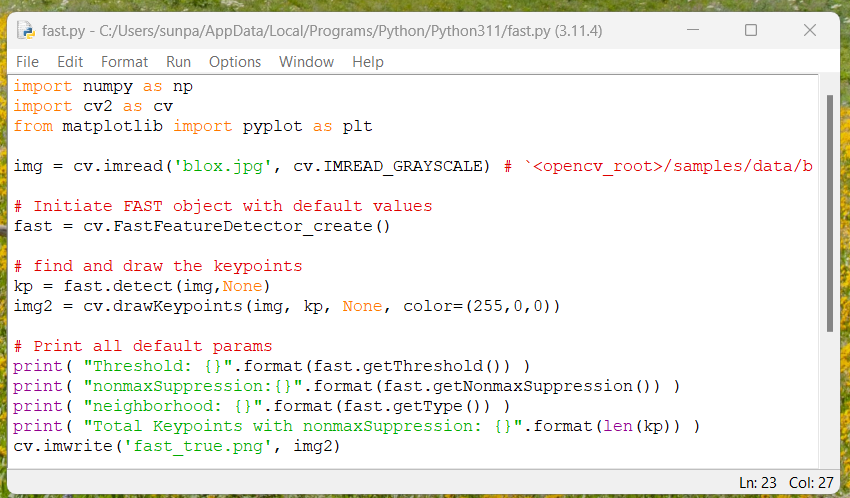
Use ID3 decision tree classifier to query each subset using variable Kp for knowledge about the true class, selects which x yields the most information about whether the candidate pixel is a corner measured by the entropy of Kp

Recursively apply to all the subsets until its entropy is zero

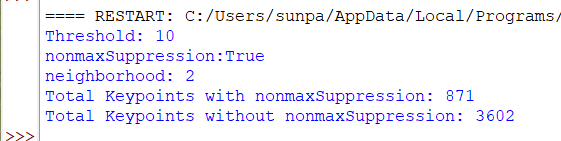
The decision tree so created is used for fast detection in other images

Non maximal suppression is used to weed out multiple features detected close together

SCore function V is the sum of the absolute difference between p and 16 surrounding pixels and values



Code with and without non maximal suppression



Results

