DIGITAL SIGNAL ANALYSIS AND APPLICATION : PROJECT

AUTOMATION OF SLIDE MATCHING

Adhithya Arun (20171066) | Meher Shashwat Nigam (20171062)

1. Introduction

These days, the demand for online lectures is increasing. For better visual experience, along with the video of the lecture, soft copy of the slides is also being embedded into the video. But most of the universities are manually matching slides from the video to the soft copy which is a laborious task. So the problem statement is to automate this slide matching process.

2. Approach

For this problem, we started off directly with ***Normalized Cross Correlation***, a method we have used earlier for template matching in one of our assignments. This method however turned out to be rather a poor measure for comparing the slides, for most of our slides and frames were mismatched. The results were far below expected and hence, we had to look for other options. The formula for normalized cross correlation is given below:

$$R(x,y) = \frac{\sum_{x',y'}(T(x',y') \cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}}$$

Following this failure, we tried to use ***Structural Similarity*** (SSIM) which was another method for template matching which does far better than MSE or Normalized Cross Correlation. The formula for calculating the structural similarity is as given below:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

The results from SSIM were promising, however, upon further searching and comparison, we found better results in other methods.

We compared SSIM to ***Cross Correlation***, ***Square Difference*** and ***Correlation Coefficient***, as well as their normalized forms. The normalized methods performed better than the former, with the best being ***Normalized Correlation Coefficient*** (or *Correlation Coefficient, both of which performed the same*). The formula used for the above mentioned methods are given below:

Normalized Correlation Coefficient

$$R(x,y) = \frac{\sum_{x',y'}(T'(x',y') \cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}}$$

Normalized Square Difference

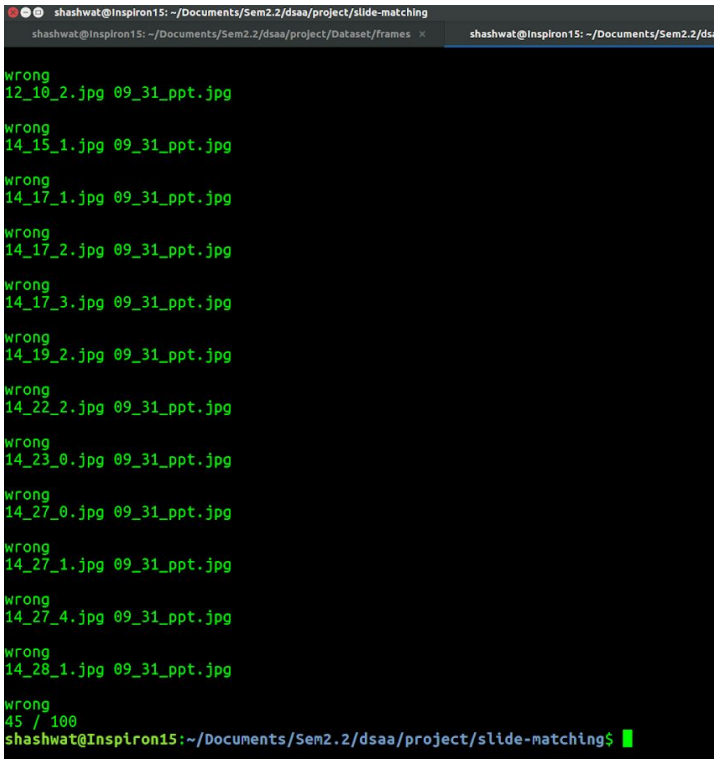
$$R(x,y) = \frac{\sum_{x',y'}(T(x',y') - I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}}$$

3. Performance Analysis

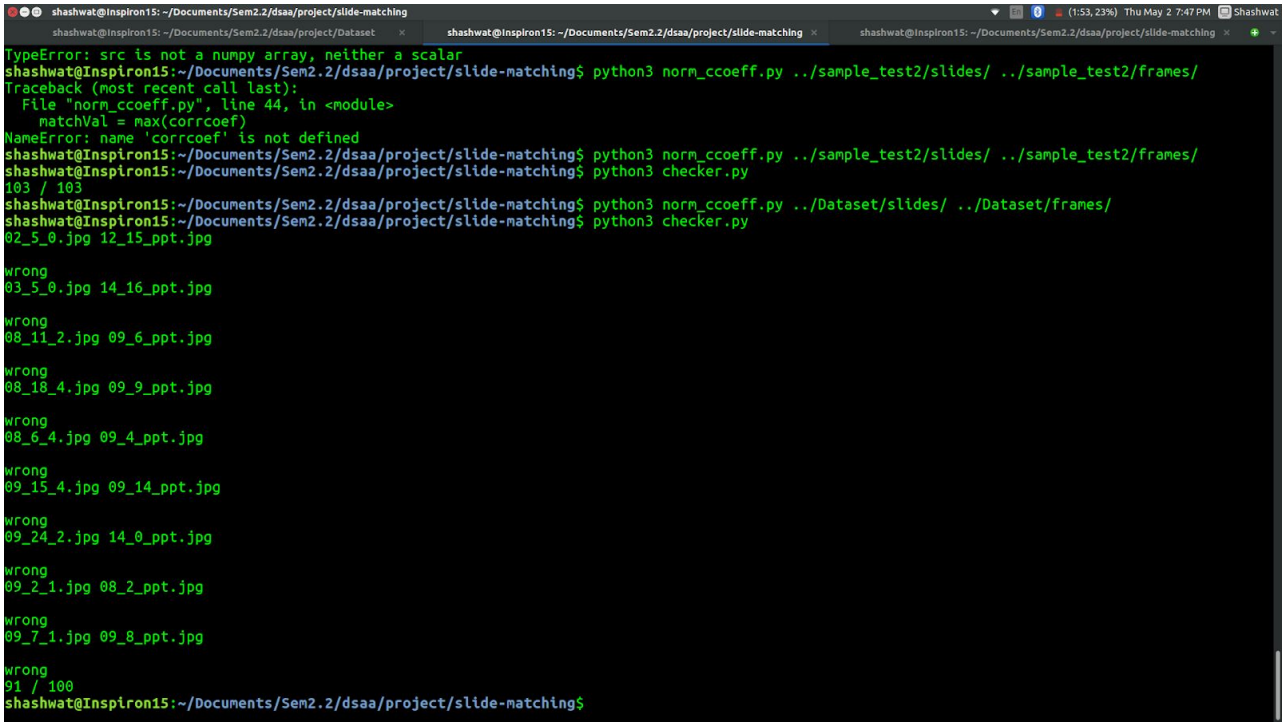
Here we present the data we collected on the tests we ran using different sizes of datasets with different methods. To finalize the best method, we compared all of the methods using 126 frames and 20 corresponding slides, and checked the accuracy rate. The results are tabulated below:

Method	Accuracy
Cross Correlation	11.90%
Normalized Cross Correlation	28.57%
Correlation Coefficient	94.44%
Normalized Correlation Coefficient	94.44%
Square Difference	41.27%
Normalized Square Difference	61.90%
Structural Similarity	81.74%

After the initial comparison, we identified Normalized Correlation Coefficient and SSIM as the best 2 methods, and did further testing on them.We created a new dataset with hundred random frames from the complete dataset ; and all the slides. We ran SSIM and Normalized Correlation Coefficient again on this dataset. The results obtained are as shown. **Normalized Correlation Coefficient was found to work much better and we are presenting this as our solution algorithm.**



With SSIM : 45% accuracy



With Normalized Correlation Coefficient : 91 %.

4. Submission

We have submitted implementations for :

- **Normalized Correlation coefficient -> *20171066_20171062.py : final submission.***
- Structural similarity(SSIM) -> *match_two_factor.py*
- Correlation coefficient -> *ccoeff.py*
- Normalized Cross Correlation -> *match.py*

Also a few auxiliary files for testing:

- *sample_maker.py* -> converts the given dataset/ subset of it into the input format specified, changes name so that checking is easier.
- *checker.py* -> parses the output file (output generated from renamed files; generated from sample_maker.py) and returns the wrong matches along with the score : (correct/total).

Output :

On running:

python3 20171066_20170162.py <path to slides> <path to frames>

20171066_20171062.txt

- is generated as output with the mapping.