**0510-6201 – Digital Signal Processing**

**Final Project**

Title

Roy Hachnochi 311120851

Uri Kapustin 317598811

**Date:** 01.07.21

**Summary of the chosen paper:**

TODO:

* Explain the paper, the math, the algorithm, and the innovation.
* Weighted Nuclear Norm Minimization with Application to Image Denoising

**Related work:**

TODO:

* backward related papers:
  + Image Denoising by Sparse 3-D Transform-Domain (BM3D)
  + From Learning Models of Natural Image Patches to Whole Image Restoration
  + A singular value thresholding algorithm for matrix completion
  + find more (take references from our original paper, e.g. nuclear norm, block matching,…)
* follow-up papers:
  + Multi-channel Weighted Nuclear Norm Minimization
  + Multi-Scale Weighted Nuclear Norm Image Restoration
  + Turning a Denoiser into a Super-Resolver Using Plug and Play Priors (?)
  + An Improved WNNM Algorithm for Image Denoising
  + Note that many papers refer to this paper as SOTA and compare their methods to its achievements.

**Our project:**

TODO:

* Explain why the straight-forward solution (taking all patches from all frames together/concatenating all frames to a single image) is bad (computational cost…)
* Explain our innovation
* related work:
  + Robust video denoising using Low rank matrix completion
  + Video Denoising by Sparse 3D Transform-Domain Collaborative Filtering
* describe the algorithm
  + block matching + predictive block searching
  + WNNM
  + patch aggregating
* describe implementation details
* show some results and examples:
  + different types of noise
  + different levels of noise
  + make a histogram of which patches were matched (from consecutive frames or from same frame, temporal/spatial)
  + make a graph showing the PSNR as a dependency of how many patches we use, or of the proportion of spatial/temporal patches in group.
  + Also check run-times for various methods/parameters.
* summary

**Code:**

* Block matching module (including consecutive frames and same image).
* Denoising module:
  + Weight calculation
  + SVD
  + SV Thresholding (iteratively solving the optimization problem).
  + Image restoration based on patches.
* Metrics (PSNR, SSIM, others?)
* Main script: examples, tests, histograms, graphs.
* Take care of RGB.