**0510-6201 – Digital Signal Processing**

**Final Project**

**WNNVD – Weighted Nuclear Norm for Video Denoising**

Roy Hachnochi 311120851

Uri Kapustin 317598811

**Date:** 01.07.21

**Table of Contents**

[1. Summary of the chosen paper 1](#_Toc80286664)

[1.1. Nuclear Norm 1](#_Toc80286665)

[1.2. Weighted Nuclear Norm 1](#_Toc80286666)

[1.3. Image denoising 1](#_Toc80286667)

[1.4. The solution 2](#_Toc80286668)

[1.5. The algorithm 3](#_Toc80286669)

[1.6. Results 3](#_Toc80286670)

[2. Related work 4](#_Toc80286671)

[3. Our project – WNNVD 5](#_Toc80286672)

[3.1. Related work 5](#_Toc80286673)

[3.2. Method 6](#_Toc80286674)

[3.2.1. Preprocessing 6](#_Toc80286675)

[3.2.2. Block Matching 7](#_Toc80286676)

[3.2.3. Group Extraction 8](#_Toc80286677)

[3.2.4. WNNM 8](#_Toc80286678)

[3.2.5. Aggregation 8](#_Toc80286679)

[3.2.6. Loop 8](#_Toc80286680)

[3.3. Implementation details 8](#_Toc80286681)

[3.4. Results 9](#_Toc80286682)

[3.5. Future work 9](#_Toc80286683)

[4. Summary 9](#_Toc80286684)

[References 10](#_Toc80286685)

# Summary of the chosen paper

In this project we chose to focus on the paper “Weighted Nuclear Norm Minimization with Application to Image Denoising” [*Gu et. al.*] (reference). The paper was published in CVPR2014 and expands the idea of Nuclear Norm minimization by adding weights to the optimization problem. The paper also proposes an algorithm for solving the optimization problem for various cases of the weight vector and demonstrates a practical use for an application of image denoising.

## Nuclear Norm

The nuclear norm of a matrix is defined as:

Where are defined as the singular values of (ordered decreasing).

## Weighted Nuclear Norm

As presented in the paper, the weighted nuclear norm of a matrix is defined as:

Where is the assigned weight of the singular value .

Obviously, this is an expansion of the Nuclear Norm, and reduces to it when given .

The weighted nuclear norm is non-convex in general, making it much harder to optimize. However, the paper shows that when are non-ascending the norm remains convex, and additionally propose methods for optimization in other, more general, cases.

## Image denoising

Based on previous papers (add references) the paper proposes the following low-rank minimization problem with a weighted nuclear norm regularization:

We will explain the terms of the above equation:

The image denoising algorithm is a non-local self-similarity (NSS) method, which takes advantage of the fact that a natural image holds many repetitions of local patches. This fact may be utilized by finding estimates of these local patches using methods such as Block Matching (reference) and using the similarity of the patches to achieve a better estimate of the original patch. We denote as the matrix in which each row is a vectorized patch of the NSS group, as the estimation of the denoised matrix, and as the noise level (variance), assumed to be known.

The optimization problem thus adds the assumption that is actually a low-rank matrix because it’s rows (patches) are similar. The first term in the optimization problem is the data-term, which aims for the estimated patches to be somewhat similar to the noised patches. The second term is the low-rank prior, aiming to achieve similar patches by forcing the matrix to be of low rank.

For the nuclear norm problem () the solution was previously (reference) shown to be:

The above is called the soft thresholding function. The problem with the solution for the non-weighted version is that it decreases each singular value by the same term, not taking advantage of the prior knowledge that the largest singular values actually hold the most information on the signal and we should thus apply a softer thresholding on them.

## The solution

The paper divides the solution under three cases regarding the weight vector:

1. The weights are in a non-ascending order:

The paper proves that for this particular case the optimization problem is still convex, and that the solution is given by:

Where is the SVD of , and is the generalized soft-thresholding operator, given by:

1. The weights are in arbitrary order:

For this case the paper proposes an iterative process for finding , also using the generalized soft-thresholding operator. We will not expand on this since this case isn’t important for our work, but only for the deduction of the last case.

1. The weights are in a non-descending order:

The paper proves that for this case the iterative process from above has a fixed point, given by the same solution of the first case:

The final case is actually the most important one, since (as stated previously) we would like to give lower weights to the larger singular values, therefore resulting in a softer thresholding for them.

## The algorithm

The Weighted Nuclear Norm Image Denoising algorithm (WNNID) presented in the paper follows the solution of the optimization problem in ‎1.4 and takes the following steps:

Text, letter

Description automatically generated

## Results

The algorithm achieved state-of-the-art results in its time, overcoming all competing algorithms for all noise levels and all test images. An example for a denoised image from the paper:

Diagram

Description automatically generated

# Related work

The paper cites several other papers, the most worth mentioning ones to our opinion (which also helped us understand the context and ideas behind the innovation of the paper) are the following:

* + “Image Denoising by Sparse 3-D Transform-Domain” (BM3D) (reference):

This algorithm was the state-of-the-art image denoising algorithm prior to our chosen paper. It is based on the same NSS principle that WNNID is based on, and also performs block-matching and aggregation. The difference is the optimization problem being solved to clean each patch-group. In BM3D the groups are denoised using collaborative filtering and Wiener filtering.

This algorithm is also used as a comparison for the WNNID algorithm. We note that the WNNID is much simpler, and also achieves better results (although it may have a higher computational cost).

The WNNID algorithm also bases the block-matching step on the block-matching proposed by BM3D. This is a very important step in the algorithm, since the success may be only as good as the quality of the matched patches.

* + “A non-local algorithm for image denoising” (reference):

Amongst other contributions, this paper presents the Nonlocal Self-Similarity (NSS) approach, which was vastly adopted in following publications, including WNNID.

* + “A singular value thresholding algorithm for matrix completion” (reference):

This paper presented the Nuclear Norm Minimization problem as a relaxation of the NP-hard rank-minimization problem and proposed the soft-thresholding operator which was expanded in the WNNID paper.

We also mention several papers which expanded the work presented in our chosen paper:

* + “Multi-Scale Weighted Nuclear Norm Image Restoration” (reference):

This paper adopted the idea of Weighted Nuclear Norm image denoising for the more general task of image restoration (e.g., deblurring, inpainting). They did so by proposing a half quadratic splitting (HQS) technique based on a generalized version of the problem presented in ‎1.3. They also added multi-scale patches (patches not only from the image itself, nut also from scaled down versions of the image), relying on previous work which showed the property of recurring patches across different scales. The paper presented competitive and state-of-the-art results for the tasks of image deblurring and image inpainting.

* “Multi-channel Weighted Nuclear Norm Minimization” (reference):

This paper proposed an algorithm called MC-WNNM (Multi-Channel Weighted Nuclear Norm Minimization), expanding the grayscale image denoising performed by WNNM to a RGB colored image denoising scheme. This is done by concatenating patches from the 3 channels to form the patch vector ( being the patch size) and formulating the following optimization problem:

Where are the grouped noise and estimated similar patches appropriately, and is a weight matrix to balance the noise levels between channels. The paper also proposes a solution for this problem, employing the variable splitting method (references) and solving using the alternating direction method of multipliers (ADMM) (references).

* + “An Improved WNNM Algorithm for Image Denoising” (reference):

This paper shows that the WNNM algorithm, while achieving state-of-the-art results for white gaussian noise denoising, attains bad performance for salt & pepper denoising. The paper solves this problem simply by performing a two-stage algorithm: firstly, using WNNM to denoise the image, and then applying adaptive median filtering to process the remaining salt & pepper noise.

* + Additionally, we also note that many other papers from the following years cite this paper as a state-of-the-art image denoising algorithm.

# Our project – WNNVD

After reading the paper and the additional related papers, and seeing the impressive results achieved for the image denoising task, we initially thought to expand the idea of using WNNM for image denoising to use it for image deblurring, an idea which was not explored in the original paper. We soon found out that (reference) dealt exactly with this problem and solved it impressively. We also thought about expanding the proposed algorithm to deal with RGB (multi-channel) images and to check the performance of the algorithm on more types of noise (Salt & Pepper, Poisson…).

Eventually we chose to expand the idea of image denoising to the video denoising task. We will describe our chosen method, note our innovation, show some results, and discuss further possible improvements.

## Method

The obvious and simplest approach would be to apply the WNNID algorithm per-frame sequentially. Except from being time-consuming, this naive method lacks a strong assumption which may be taken in advantage when dealing with videos. This assumption is that there is a high temporal familiarity in videos, meaning that many patches are repeated between neighboring frames. For example, the background stays almost exactly the same, or take a moving ball which looks the same between neighboring frames but only slightly changes its location.

This important assumption is what drives our innovation – we improve the block matching phase of the algorithm to find not only spatially similar patches, but also search in the temporal dimension. This will result in two advantages. The first is that we now base our WNNM phase on more/better matched patches in each group. The second is that with this method we process patches from several frames at once, sparing the need to process each frame individually and thus allowing a lower computational cost.

Given a noised video , our model may be described using the following block diagram:

A screenshot of a computer

Description automatically generated with medium confidence

### Preprocessing

As suggested in many image/ video denoising algorithms based on the concept of block matching, including (references), we perform a preprocessing phase which includes a naive per-frame denoising method, e.g., Gaussian filtering, Median filtering. The denoising helps in the block matching process by making patches which were originally similar but were contaminated by noise, become similar again, thus making them easier to compare between. Note that the chosen denoising methods may be changed orthogonally to the WNNVD algorithm to match specific priors on the video.

### Block Matching

While this part of the algorithm is not the true innovation of the original paper, this may be considered the most crucial one since low-quality block matching will result in un-correlated groups and thus in a bad patch-restoration when applying the WNNM method. Furthermore, this is where our true innovation lies.

As explained in ‎3.2, we wish to take advantage of the temporal correlation between patches in addition to the original spatial correlation suggested in (reference). We therefore adapt the video block matching method proposed in (reference), which may be described as follows for each of reference patches described by the indices :

Non-predictive block matching:

We search for patches in the current frame which are similar to the reference patch by a brute-force search in a window around the reference patch indices, using a stride of to reduce computational cost. Distances for all patches in this window are calculated, and the most similar patches are chosen. This results in the matrix of nearest patch indices in the current frame, and the vector holding their distances.

Predictive block matching:

For each ( being the temporal search window, and excluding the frame and corner cases of the video where or ), we take the matched patches from the previous frame ( if and if ) and open search windows around each patch index, with a search window of size and stride . Note that may be significantly smaller than since we have already found the spatially similar patches and now should only find their movement between neighboring frames. Once again, distances for all patches in all search windows are calculated, and the most similar patches are chosen, resulting in and .

After finding and for all , the most similar patches are chosen for the group, resulting in the matrix .

The non-predictive and predictive searches are performed for each reference patch, ultimately resulting in the matrix .

Below is an example for predictive block matching through 3 following frames:

A picture containing text, flag

Description automatically generated

Frame #10 is the reference frame, the red rectangle is the reference patch, the green rectangles represent the search window (notice the larger window for the predictive BM in frame #10, and the multiple smaller windows for the following frames), the cyan rectangles are the matched patches per frame, which are also the indices for the search windows for the following frame. Notice how even though the wheel moves a bit between frames, our predictive block matching process allows finding the moved patches. Also notice that there appear to be matched patches outside of a legal search window. This is not a bug but is caused by the fact that we choose the globally most similar patches only after the block matching is done for all frames, causing some of the actual search windows to disappear from our plot.

### Group Extraction

For each of the groups found in ‎3.2.2, the patches are extracted from the video, vectorized, and concatenated to form the matrix (where denotes the patch size).

### WNNM

This step is actually identical to that of the original WNNID algorithm, since it takes a group matrix , blind to whether the patches were extracted from the same frame or different frames. This step performs the WNNM thresholding algorithm and returns , by applying the generalized soft-thresholding operator as presented in ‎1.4.

### Aggregation

Two matrices are formed in this step: and . After denoising the patch group and obtaining , each denoised patch is then placed back in its corresponding position using . Since different patches may overlap, we actually sum all patches to , and hold a count matrix for each pixel . After placing all denoised patches, is obtained by dividing pixel-wise. Pixels which were not aggregated in this step are simply taken from the original video .

### Loop

Since the above steps are performed using a single reference frame , and since some frames/patches may remain un-processed after these steps due to not reaching the frame on not grouping the patches, we perform steps ‎3.2.2-‎3.2.5 multiple times, each time choosing a different reference frame based on the frame with the least processed pixels. Note that the algorithm is guaranteed to end since for each reference frame, all its pixels are automatically marked as processed since they all appear in reference patches.

## Implementation details

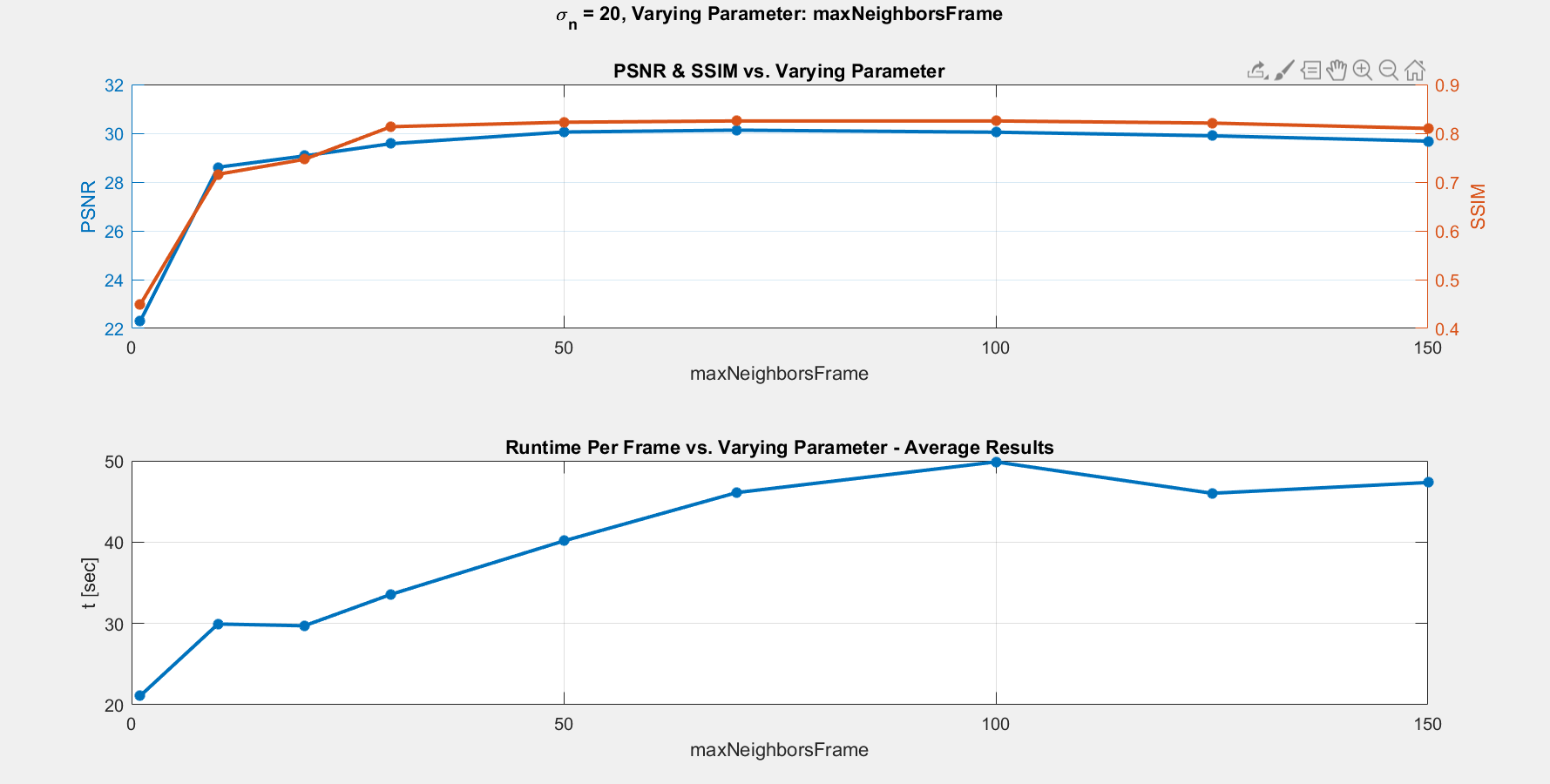
TODO

## Results

* Show table comparing WNNID, VBM3D, WNNVD.
* Show example of block-matching, including histogram between frames.
* Maybe – compare block matching using only temporal blocks to using a spatio-temporal mix.
* Maybe – 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.
* Show results on an image (lena/single frame) to show that our algorithm works well on a single image.
* Show some examples of frames from different videos.

Graphical user interface, application

Description automatically generated



blue: gbicycle

green:gflower

red: gmissa

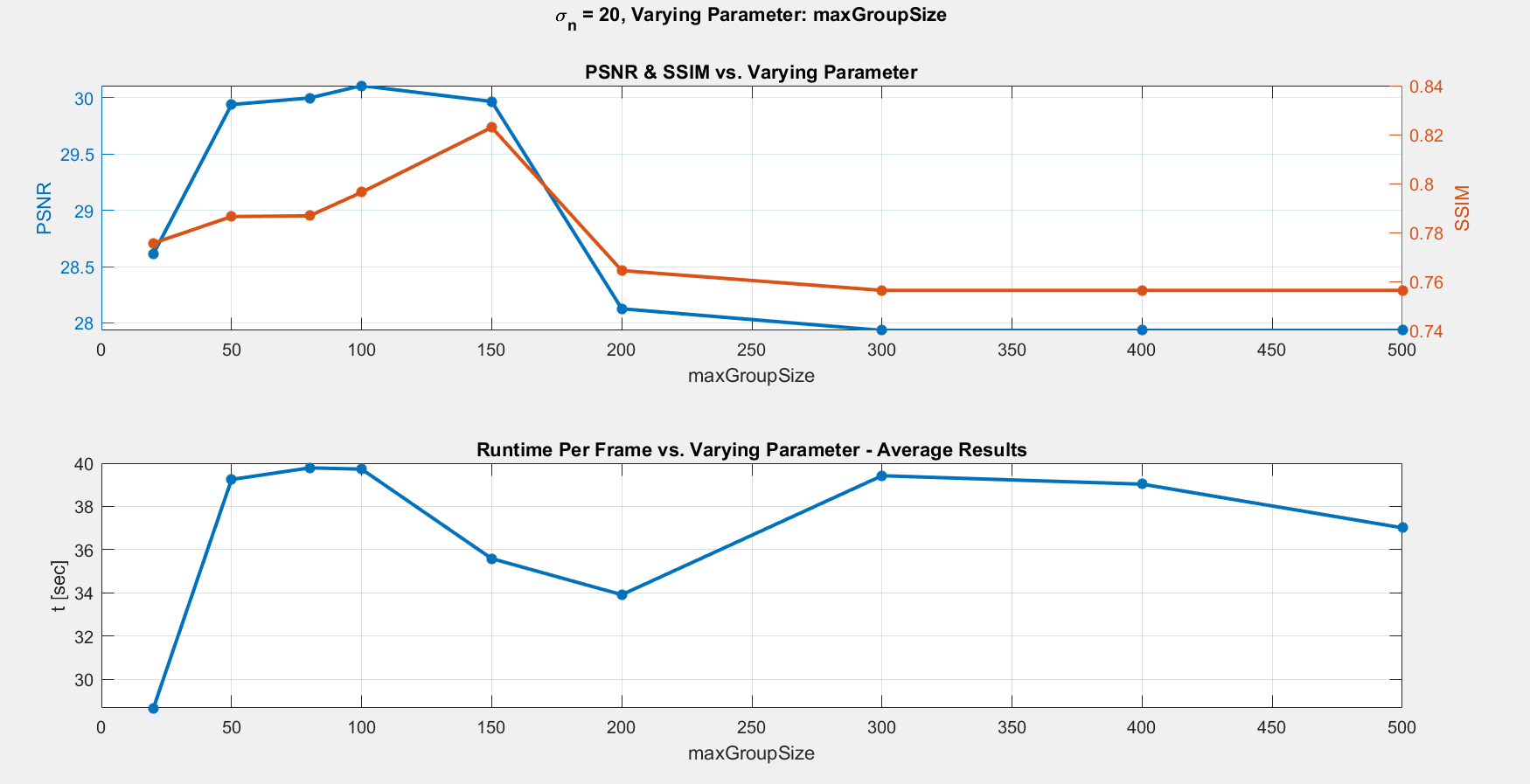
cyan: gsalesman

magenta: gstennis

yellow:lena

Graphical user interface, chart

Description automatically generated



Graphical user interface, chart

Description automatically generated

Chart, line chart

Description automatically generated

## Future work

TODO

# Summary

TODO

# References

|  |  |
| --- | --- |
| [1] | C. Manning, "Stanford, CS224n: Natural Language Processing with Deep Learning," [Online]. Available: http://web.stanford.edu/class/cs224n/ . |
| [2] | M. K. e. al., "Overview of the Author Identiﬁcation Task at PAN 2018 - Cross-domain Authorship Attribution and Style Change Detection," in *PAN*, Avignon, 2018. |
| [3] | M. K. e. al., "Overview of the Cross-Domain Authorship Attribution Task at PAN 2019," in *PAN*, Lugano, 2019. |
| [4] | Y. Sari, "Neural and Non-neural Approaches to Authorship Attribution - A Thesis submitted to the University of Sheﬃeld for the degree of Doctor of Philosophy in the Faculty of Engineering," Department of Computer Science The University of Sheﬃeld, 2018. |
| [5] | J. S. S. A. Moshe Koppel, "Computational Methods Authorship Attribution," *Journal of the American Society for Information Science and Technology,* vol. 60, no. 1, pp. 9-26, 2009. |
| [6] | E. Stamatatos, "A Survey of Modern Authorship Attribution Methods," *Journal of the American Society for Information Science and Technology,* vol. 60, no. 3, pp. 538-556, 2009. |
| [7] | A. Karpathy, "The Unreasonable Effectiveness of Recurrent Neural Networks," 21 May 2015. [Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness/. |
| [8] | T. Tran, "Text Generation With Pytorch," 08 Feb 2019. [Online]. Available: https://machinetalk.org/2019/02/08/text-generation-with-pytorch/. |
| [9] | "Project Guttenberg," [Online]. Available: https://www.gutenberg.org/. |
| [10] | "The British Medical Journal," BMJ Publishing Group Ltd, [Online]. Available: https://www.bmj.com/. |
| [11] | S. E. L. a. E. K. Bird, *Natural Language Processing with Python,* O’Reilly Media Inc, 2009. |
| [12] | S. a. S. J. Hochreiter, "Long Short-Term Memory," *MIT Press,* vol. 9, no. 0899-7667, 1997. |
| [13] | T. W. e. al., "HuggingFace's Transformers: State-of-the-art Natural Language Processing," *ArXiv,* vol. abs/1910.03771, 2019. |
| [14] | M. Ankerst, M. M. Breunig, H.-P. Kriegel and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," *ACM SIGMOD international conference on Management of data. ACM Press,* pp. 49-60, 1999. |
| [15] | L. van der Maaten and G. Hinton, "Visualizing Data Using t-SNE," *Journal of Machine Learning Research,* no. 9, p. 2579–2605, Nov 2008. |
| [16] | I. e. a. Kononenko, "Overcoming the myopia of inductive learning algorithms with RELIEFF," *Applied Intelligence,* vol. 1, no. 7, pp. 39-55, 1997. |