University POLITEHNICA of Bucharest

Faculty of Automatic Control and Computers,
Computer Science and Engineering Department

PHD THESIS SUMMARY

On Selecting the Best Approach: From Image Analysis and Processing to Software Development

Scientific Adviser:

Prof. Phd. Eng. Costin-Anton Boiangiu

Author:

Cristian-Dumitru Avatavului

Bucharest, 2023

Contents

1 Introduction

In the milieu of digital transformations in contemporary society, image analysis and the prediction of physical interactions have emerged as increasingly relevant and complex areas of research[1]. These domains are pivotal for the advancement of cutting-edge technologies and the enhancement of human-machine interaction across various applications, ranging from visual data processing to complex simulations of physical interactions [2].

1.1 Organization and Structure of the Thesis

The architecture of this doctoral thesis is designed with the aim of providing a systematic and coherent analysis of the subjects discussed. Its objective is to facilitate the understanding and monitoring of research progress.

Chapter 1 provides an expansive context for the research, emphasizing the significance of digital transformations in modern society, the importance of image analysis, and the prediction of physical interactions in the development of advanced technologies and the improvement of human-machine interaction. The purpose and objectives of the work are clearly delineated in this section, outlining the major goals of the thesis.

Chapter 2 constitutes the first pillar of the research and delves into key aspects of image analysis, such as Optical Character Recognition (OCR), motion analysis, super-resolution, page tilt correction, and contrast assessment.

Chapter 3 aims to identify and group relevant entities into a hierarchical structure, thereby offering a comprehensive perspective on the organization of document pages.

Chapter 4 is dedicated to exploring the potential of neural networks in predicting dynamic physical interactions. By applying these advanced technologies, complex phenomena that occur in different contexts are better investigated and understood, contributing to a deeper understanding and simulation of physical interactions.

Chapter 5 addresses essential aspects, comparing development paradigms of "Indie versus AAA" and "open-source versus closed-source." Through comparative analysis, the most efficient software development strategies are identified, considering costs, quality, and industry impact.

Chapter 6 serves as a synthesis of the research outcomes, underscoring the contributions made in the fields of image analysis, prediction of physical interactions, and efficient software development.

2 Image Analysis Through Efficient Combination of Results from Multiple Algorithms

In this chapter, we focus on image analysis and innovative methods to combine the results garnered from multiple algorithms to enhance performance and accuracy in various aspects of image processing. The following subsections elaborate on specific topics within image analysis, such as Optical Character Recognition (OCR), motion analysis, super-resolution, page tilt correction, and contrast assessment.

2.1 Optical Character Recognition (OCR)

We live in a technologically advanced era, where a vast number of physical documents have been or need to be digitized to make them universally accessible for a broad audience. Optical Character Recognition (OCR) is one of the widely used techniques to recognize characters from specific images acquired through scanning. Various types of systems have been developed to perform optical character recognition in diverse types of documents. However, the task is not straightforward, as documents differ not only in terms of content but also in formats, fonts, age, and deterioration. After reviewing existing systems, we propose one that employs two well-known OCR engines and a weighted voting principle. The results of our combined technique are analyzed and contrasted against the individual approaches of the two selected engines [3].

Most widely used OCR engines:

Ocropy [1] (also known as Ocropus) is an OCR engine based on Long Short-Term Memory (LSTM). It tackles the complex challenges of character recognition in images with diverse fonts and varying image quality [4].

Ocrad [2] (also known as GNU OCR) is an optical character recognition technique developed within the GNU Project (GNU's Not Unix). It offers an open-source solution for transforming text images into digital content, enabling users to automatically extract and process information from images [5].

[1] https://en.wikipedia.org/wiki/OCRopus

[2] https://www.gnu.org/software/ocrad/manual/ocrad_manual.html

SwiftOCR[1] **-** is an optical character recognition (OCR) library specialized for Apple platforms such as iOS and macOS. Developed in the Swift programming language, the library seamlessly adapts to the Apple ecosystem, offering native integration with services and technologies like Core Image and Vision Framework [6].

Attention-OCR[2] **-** represents a modern approach in the field of optical character recognition, based on machine learning techniques, and more precisely, on neural network models with an attention mechanism [7].

Tesseract[3] - is an OCR project developed by Google since 2006 [8]. It has evolved significantly over the years, moving from a simple text reader based on Neural Networks (NN) without template analysis support, to a comprehensive system that recognizes common templates and offers support for both NN and LSTM recognition. Subsequent versions of Tesseract support various output formats, including hOCR (HTML-based OCR) with layout and formatting information, and can be integrated with user interfaces such as Ocropus [9] [10].

Asprise[4] - is a powerful and reliable OCR engine developed by Asprise. It offers advanced character recognition capabilities and allows users to access OCR functionalities through a simple Application Programming Interface (API). Asprise OCR stands out for its ability to handle images of various qualities and formats, as well as for its performance in recognizing characters in complex texts [5].

Previously, we introduced a range of modern and high-performance OCR engines. Nonetheless, Tesseract and Asprise have been selected due to key factors: maturity, community support, functionality diversity, documentation, support, as well as performance and accuracy.

2.1.1 Method of Combining Results from Tesseract and Asprise

Within the scope of this study, an analysis of the OCR engines Tesseract and Asprise was conducted, followed by the proposal of a new system that combines both technologies. This system implements a weighted voting mechanism to achieve the most accurate possible result.

[1] https://www.swift.org/

[2] https://github.com/emedvedev/attention-ocr

[3] https://tesseract-ocr.github.io/tessdoc

[4] http://asprise.com

The input file processing procedure is divided into a series of steps, as can be seen in figure 2.1 [3] [11].

By applying this weighted voting mechanism, the new system ensures a more precise and robust approach in the optical character recognition process, thus leading to an improvement in the quality and accuracy of the final results. The outcomes obtained through this new approach will be scrutinized and compared in detail to evaluate the performance and efficiency of the proposed system relative to traditional methods. Ultimately, this study will contribute to the advancement and improvement of OCR technologies, yielding significant benefits across various application domains [12].

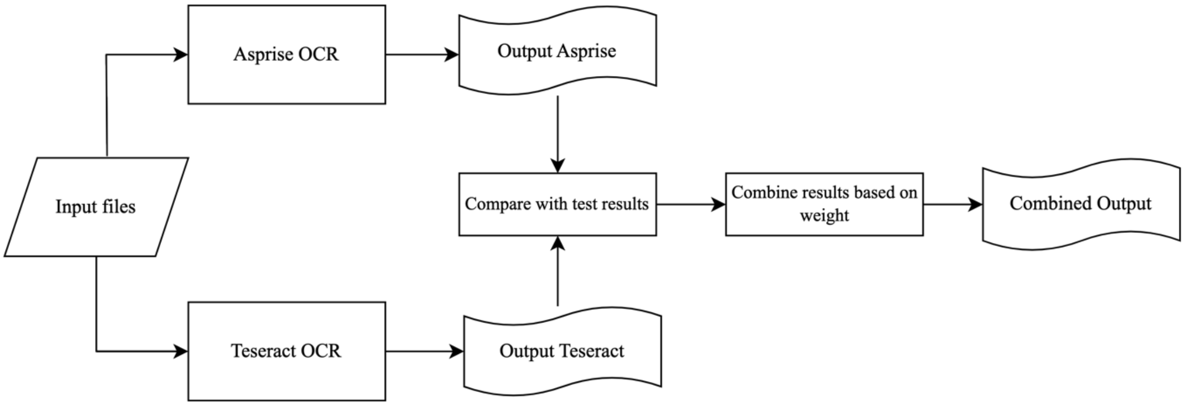


Figure 1.1: Steps for processing input files [3].

Upon receiving the input file, each OCR engine creates its own visual interpretation of the text based on its confidence level. The input file is represented in the first image of figure 2.3, and the OCR outputs are delineated in subsequent images, color-coded according to the confidence level specific to each technique.

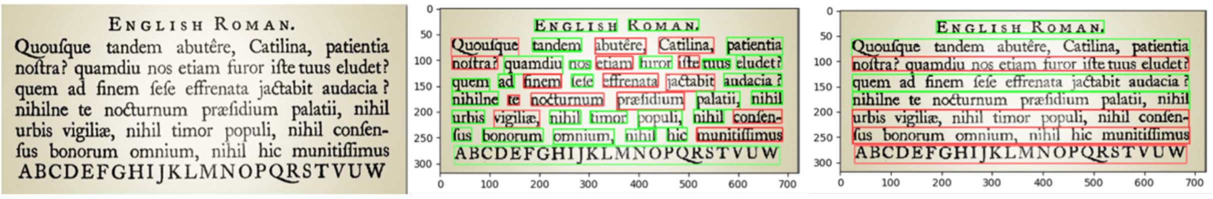
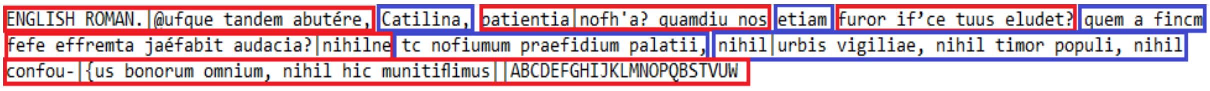


Figure 1.2: Test image and the visual result generated by Tesseract and Asprise [3].

In the final stage, we combined the documents based on the assigned weights and obtained the final result, which can be observed in figure 2.6.



```
ENGLISH ROMAN.|@ufque tandem abutére,|Catilina,|patientia|nofh'a? quamdiu nos|etiam|furor if'ce tuus eludet?|quem a fincm
fefe effremta jaéfabit audacia?|nihilne|tc nofiumum praefidium palatii,|nihil|urbis vigiliae, nihil timor populi, nihil
confou-|{us bonorum omnium, nihil hic munitiflimus||ABCDEFGHIJKLMNOPQBSTVUW
```

Figure 1.3: Results after combination (Tesseract in red and Asprise in blue) [3].

A series of experiments were conducted to evaluate the most appropriate weights. Figure 2.7 displays some results according to the confidence level of each OCR engine. The color intensity represents the level of confidence: green means high confidence, red means low confidence.

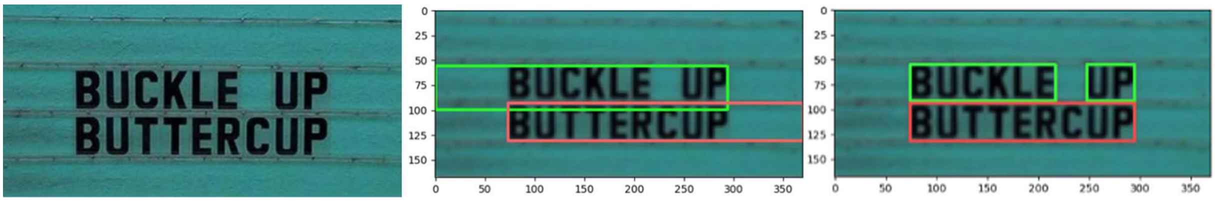


Figure 1.4: Test image, Tesseract result, and Asprise result [3].

After analyzing multiple scenarios, it was observed that, at any level of image blurring, Asprise tends to perform better than Tesseract, a difference evident in Figure 2.8.

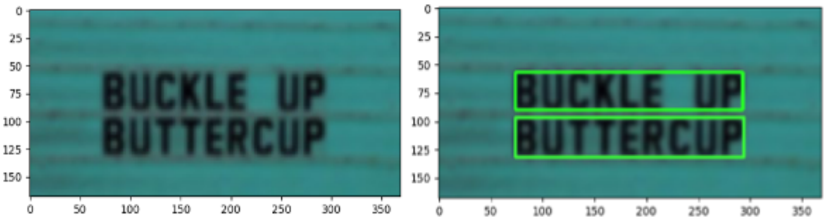


Figure 1.5: Tesseract and Asprise outputs [3].

Comparing the results, it is evident that each engine has its own shortcomings in handling degraded input files. To minimize the potential for erroneous results from low-quality files—a challenge most engines will face—we decided to create a weighted voting mechanism that analyzes both outcomes and generates an output based on the individual confidence levels [3].

2.2 Motion Estimation and Analysis

Motion estimation is a critical topic in the field of computer vision, with a multitude of applications such as video tracking and compression. The fundamental principle of motion estimation is, given a sequence of images (typically, a video), to identify the movement, i.e., what moved in the sequence and how or how much it moved (direction and magnitude) as we traverse the images in the sequence. Typically, we aim to identify the motion between every two consecutive frames at the same point in the image space [13].

In recent years, the field of motion estimation has evolved considerably [14,15]. The literature mentions two major categories of methods for motion estimation: direct and indirect methods [16].

In the first category, we have methods presented in the scientific world:

- dense/direct methods: estimate motion in each pixel based on the temporal and spatial variation of intensity at that point [17] [18] [19];
- block matching methods: divide the image space into blocks of equal size and estimate the motion for each block based on the correlation between the pixel intensities in that block and another frame. These can be generalized into so-called "region matching" methods that use regions of arbitrary shapes instead of rectangular blocks [20–22];
- phase correlation: estimates the global motion between two frames based on their Fourier transformation. In some research materials, these are referred to as frequency domain methods [23];
- optical flow: calculates a motion vector between two consecutive frames [24].

The second category is represented by indirect methods based on features (corner detection):

- feature-based methods: these methods identify distinctive points and track them across multiple images; they estimate the motion of only the feature points based on the locations where they are found in images [25];
- dense methods rely on a Taylor series approximation valid only for small displacements of the same object point in the two frames. Thus, they are suitable for videos with a "small" amount of motion. On the other hand, block matching methods can detect more significant movements depending on the size of the search region; however, they are more computationally demanding.

The proposed method calculates the motion vector fields at the same "level of sparsity" using the two methods described above and then combines their results into a single output.

Optical Flow

This method [26,27] considers an image sequence represented by a function $I = I(x, y, t)$, where x and y are spatial coordinates (pixel locations), and t represents time (or frame number). The goal is to estimate the motion vector $[u, v]$ for each sampling point (x, y) in image space and between every two consecutive frames (t and $t + 1$).

Considering a particular point (the blue points in figure 2.9) at coordinate (x, y) in frame t, moving along the motion vector $[u, v]$, so in frame $t + 1$, the blue point will be at location $(x + u, y + v)$. A reasonable assumption is that the point will appear in the same color (or with the same intensity value) in frame $t + 1$ at the new location, providing the brightness constancy constraint (equation (1)).

$$I(x, y, t) = I(x + u, y + v, t + 1) \tag{1}$$

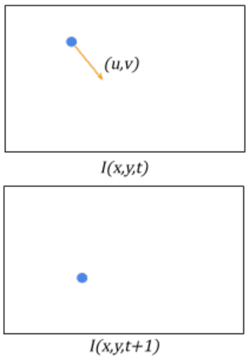


Figure 1.6: Optical Flow Example [13].

If the motion vector $[u, v]$ is small enough, the following Taylor series expansion applies:

$$I(x + u, y + v, t + 1) \simeq I(x, y, t) + \frac{\partial I}{\partial x} \cdot u + \frac{\partial I}{\partial y} \cdot v + \frac{\partial I}{\partial t} \tag{2}$$

From equations (1) and (2), we obtain:

$$I_x \cdot u + I_y \cdot v + I_t = 0 \tag{3}$$

In equations (2) and (3), $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, $\frac{\partial I}{\partial t}$, I_x, I_y, I_t represent the estimated derivatives of the image function I (în raport cu x, y, și t).

The motion vector at (x, y) can be calculated by solving equation (3) fo u and v. In this case, a unique solution is not available, and some approaches have been devised to address this:

Horn & Schunck [19]– impose a smoothness constraint on the motion vector field across the entire image; along with (II.3), this becomes a minimization problem: the motion vector field $u(x, y), v(x, y)$ must be found such that the following expression is minimized:

$$\iint (I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2) dxdy \tag{4}$$

where u_x, u_y, v_x, v_y represent the derivatives of the motion components and λ is an adjustable parameter.

Lucas & Kanade [26] – assume local smoothness of the motion vector field near the point (x, y); specifically, the assumption is that the motion vector $[u, v]$ is the same for each pixel in a small window around that point; the size of this window is a modifiable parameter. This is the method used in the voting-based algorithm presented.

Writing (3) for each pixel i in the window, we get an over-constrained system of

N equations (the number of pixels in the window) with only 2 unknowns, u and v:

$$\underbrace{\begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ \vdots & \vdots \\ I_{x_N} & I_{y_N} \end{bmatrix}}_{A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} I_{t_1} \\ I_{t_2} \\ \vdots \\ I_{t_N} \end{bmatrix}}_{b} \tag{5}$$

where the index represents the pixel number. Making the above notation, we get:

$$A \cdot \begin{bmatrix} u \\ v \end{bmatrix} = b \tag{6}$$

Above, we have an over-constrained system of N equations and 2 unknowns and, in general, it has no solution. However, we can find the "closest solution," the one that minimizes the error:

$$E = \left\| A \begin{bmatrix} u \\ v \end{bmatrix} - b \right\|^2 \tag{7}$$

by solving the normal equation:

$$(A^T A) \begin{bmatrix} u \\ v \end{bmatrix} = A^T b \tag{8}$$

If $A^T A$ is invertible, we have:

$$\begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} A^T b \tag{9}$$

which provides the following final expressions:

$$\begin{cases} u = \dfrac{\sum I_y I_t \cdot \sum I_x I_y - \sum I_x I_t \cdot \sum I_y^2}{\sum I_x^2 \cdot \sum I_y^2 - \left(\sum I_x I_y\right)^2} \\ v = \dfrac{\sum I_x I_t \cdot \sum I_x I_y - \sum I_y I_t \cdot \sum I_x^2}{\sum I_x^2 \cdot \sum I_y^2 - \left(\sum I_x I_y\right)^2} \end{cases} \tag{10}$$

where the sums are performed over the entire window around the point (x, y).

If the point (x, y) corresponds to a flat region, then the motion vector cannot be calculated because the matrix.

$$M = A^T A = \begin{bmatrix} I_x^2(x,\ y) & I_x(x,\ y) I_y(x,\ y) \\ I_x(x,\ y) I_y(x,\ y) & I_y^2(x,\ y) \end{bmatrix} \tag{11}$$

From (10), it can be seen that the complexity of calculating the value of a motion vector for 1 frame is $O(W^2)$, where W is the size of the window over which the sums are made.

Block Matching

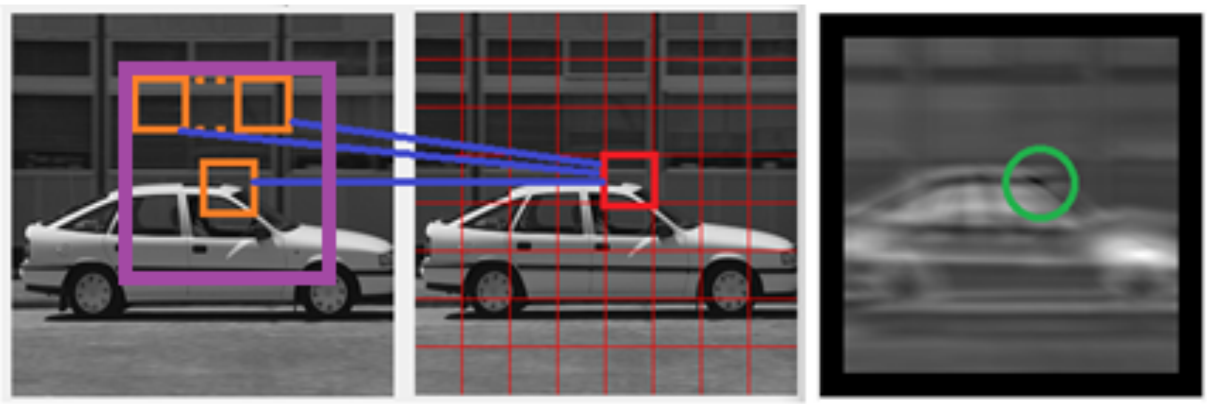


Figure 1.7: From left to right: a. reference frame; b. current frame; c. (dis)similarity function. Where: red - current block, orange - sliding window, purple - search region, green - minimum of the dissimilarity function [13].

The image is divided at time t (cadru curent) într-o grilă dreptunghiulară de blocuri, așa cum se vede în figure 2.10 (current frame) into a rectangular grid of blocks, as seen in Figure 2.10. Then, for each block, the aim is to find the location in the image space where that block was in the previous frame $t-1$ (reference frame) - i.e., the location that best matches the current block [28]. To do this, a sliding window (having the same size as the block) was considered that sweeps the reference frame and, for each location in the reference frame, how similar the region in the sliding window is to the current block was calculated. For this, a (dis)similarity function like mean squared error was used.

Thus, for a block at coordinates (x, y), and a certain position of the sliding window, offset by (d_x, d_y) from (x, y), the dissimilarity/cost function of the two regions is:

$$\epsilon(d_x, d_y) = \sum_{i,j} \left(I_t(x+i, y+j) - I_{t-1}(x+d_x+i, y+d_y+j) \right)^2 \qquad (12)$$

where the sum is performed over a region the size of the block around point (x, y). The greater the difference between the intensities of the two regions at corresponding locations, the greater the cost function. Thus, the challenge is to find the displacement (d_x, d_y) that minimizes the cost function [29]. To do this, the motion vector (u, v) is chosen to be:

$$(u, v) = -argmin_{(d_x, d_y)}\, \epsilon(d_x, d_y) \qquad (13)$$

Alternatively, a similarity function, such as cross-correlation, can be used, and we try to maximize it:

$$\epsilon(d_x, d_y) = \sum_{i,j} I_t(x+i, y+j)I_{t-1}(x+d_x+i, y+d_y+j) \tag{14}$$

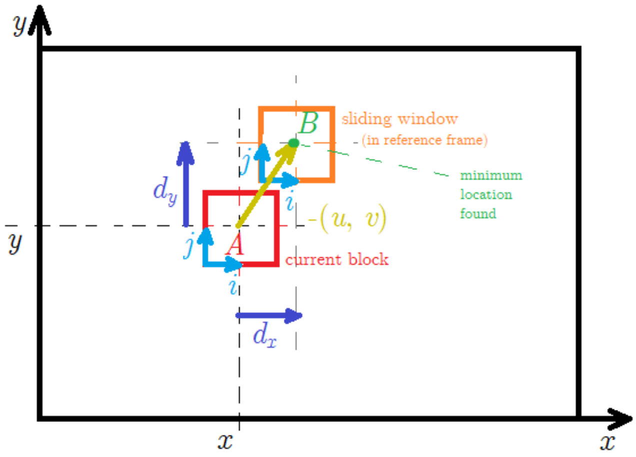


Figure 1.8: The motion compensation vector $\overrightarrow{AB}$ [13].

If A is denoted as the center of the current block and B as the location that gives the minimum cost, it can be seen that the point has moved from B to A, so the motion vector at point A is $\overrightarrow{BA}$ (figure 2.11):

$$\begin{bmatrix} u \\ v \end{bmatrix} = \overrightarrow{BA} = \begin{bmatrix} x_A - x_B \\ y_A - y_B \end{bmatrix} \tag{15}$$

If it is known that the motion is smaller than a specific value u_{max} pixels in the x, d_x can be restricted to vary in the range $[-u_{max}, u_{max}]$ (similarly for the y-direction; thus, the restriction is to search for a block in a search region around(x, y)).

The complexity of the method is $O(S^2B^2)$ for 1 motion vector and for 1 single frame, where S is the size of the search region and B is the size of the block.

Voting Method

This section presents the result of the previous two methods with (u_1, v_1) and (u_2, v_2). Then, the voting method considers the result of the voting method to be the arithmetic average of the two:

$$(u_{voting1}, v_{voting1}) = \left(\frac{u_1 + u_2}{2}, \frac{v_1 + v_2}{2}\right) \tag{16}$$

Another possibility is to take the average of both the magnitude and the angle, as seen in figure 2.12.

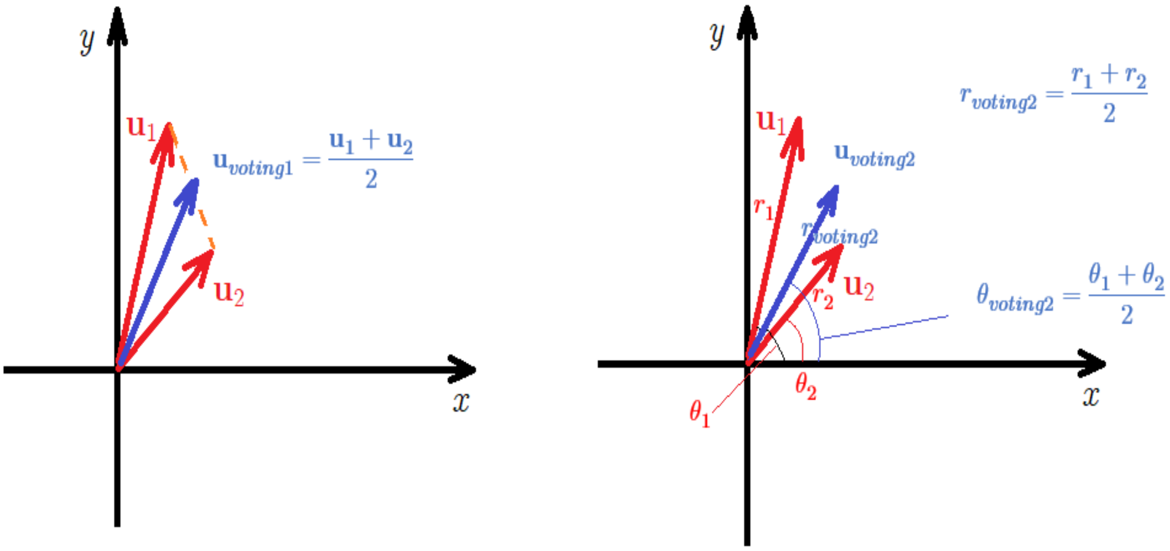

y
$\mathbf{u}_1$
$\mathbf{u}_{voting1} = \frac{\mathbf{u}_1 + \mathbf{u}_2}{2}$
$\mathbf{u}_2$
x
y
$r_{voting2} = \frac{r_1 + r_2}{2}$
$\mathbf{u}_1$
$\mathbf{u}_{voting2}$
r_1
$r_{voting2}$
r_2
$\mathbf{u}_2$
$\theta_{voting2} = \frac{\theta_1 + \theta_2}{2}$
θ_1
θ_2
x

Figure 1.9: The two cases for the proposed voting method [13].

The application opens a video and then goes through each frame, generates and displays a motion field between the current frame and the previous one. In this case, this application considered a rectangular grid over the image space and performs motion estimation at the center of each grid cell. The results of the voting approach are displayed in Figure 2.13.

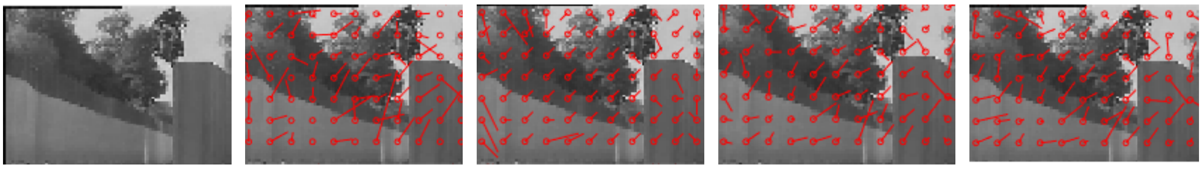


Figure 1.10: The proposed voting method. From left to right: a. the original image, b. optical flow, c. block matching, d. voting approach 1, e. voting approach 2 [13].

The methods presented are computationally intensive and have only worked for very small input images. The optical flow method often provides very imprecise results, especially at object boundaries, where the constant motion field assumption is violated. The optical flow method is unstable in flat regions (the denominator in (10) tends towards zero because the matrix in equation (11) is singular). Furthermore, this method is susceptible to noise and works only for very small motions (up to a few pixels). The block matching method is more robust than the previous one, but unfortunately, the complexity is higher.

2.3 Strategies for Image Super-Resolution

Super-resolution techniques serve as a powerful modality in the realm of image processing, aimed predominantly at enhancing the quality of low-resolution images. These methodologies find extensive applicability across a diverse range of fields, including but not limited to digital photography, video processing, medical imaging, and other applications where image quality is paramount [30] [31].

The objective of super-resolution is to transform a low-resolution image into its high-resolution counterpart by supplementing it with approximated information derived from the lower-quality version. Machine learning algorithms, particularly deep learning techniques, are aptly suited for this problem, offering superior approximations compared to traditional heuristic approaches. Deep learning algorithms, such as Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs), excel in extracting intricate and nuanced details from images [31].

In this chapter, we focus on the comparative analysis of well-established algorithms in the super-resolution domain to yield enhanced outcomes over individual algorithmic applications. The following subsections will elucidate three deep neural networks employed in our experiments:

- Residual Dense Network (RDN)
- Super-Resolution Generative Adversarial Network (SRGAN)
- Enhanced Super-Resolution Generative Adversarial Network (ESRGAN)

The algorithms selected for ensemble voting are:

- Mean Voting (pixel-based)
- Maximum Displacement - calculates the distances between pixel values for each algorithm and selects the one with maximum distance.
- Noise Estimation - the image is converted to grayscale, and noise levels are estimated; the image with the least noise (fragment-based) is selected.

For each high-resolution image, a degraded version is also provided, generated through one of the following procedures:

- Bicubic Interpolation
- Unknown Operator – the procedures generating the low-quality image are kept confidential to circumvent methods excelling solely in bicubic interpolation.

Evaluation Metrics

To gauge the quality of the images, we employ the most used metric in specialized publications and compare it with the outcomes achieved through individual deep learning algorithms. The efficacy of a network is measured by its ability to minimize the Mean Squared Error (MSE) between the output pixels and the original version. A zero MSE score indicates an identical high-resolution image generated by the network.

$$\mathrm{M}SE = \frac{1}{N}\sum_{i=1}^{N}\left(I_i - I_{\hat{(i)}}\right)^2 \tag{17}$$

$$PSNR = 10 * log_{10}\left(\frac{L^2}{MSE}\right) \tag{18}$$

The purpose of the Peak Signal-to-Noise Ratio (PSNR) is to compute the trade-off between MSE and the maximum pixel value. A higher PSNR indicates higher quality generated images.

Algoritm	**PSNR**
RDN	19.17
SRGAN	19.36
ESRGAN	19.85
Average Voting	18.98
Furthest away	19.09
Noise estimation	19.37

Table 1.1: PSNR values obtained using individual methods and proposed voting systems [31]

In this article, we tested various deep learning models that solve the problem of super-resolution and we used voting algorithms to improve their performance. Since many models used for super-resolution achieve high PSNR, but they generated images still lacking for the human eye, we can conclude that this problem has a visual perception component, represented by the subjective evaluation of human observers. Since a perfect recreation of the downscaled image is unlikely, it is necessary to take into consideration the human perspective of the results, since for practical purposes, it is useless for an image to have high PSNR and low subjective quality (as determined by human observers). Thus, results should always be correlated with human observations.

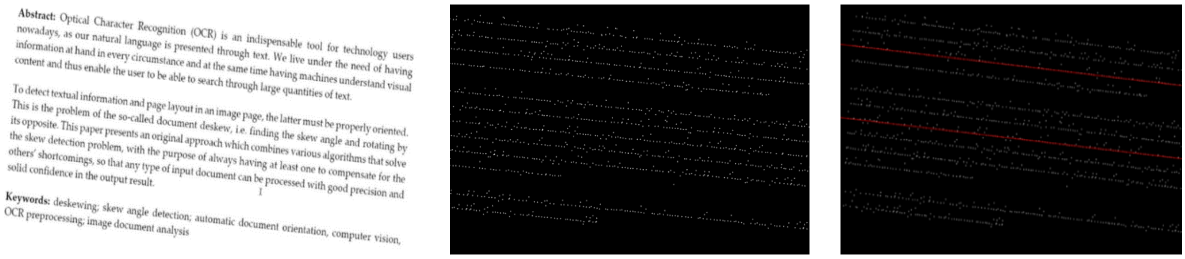
2.4 Page Skew Correction and Adaptation

Page skew correction is a critical technique in the field of image processing and Optical Character Recognition (OCR). It involves the automatic detection and rectification of skew or perspective distortions in a scanned page image, thereby aligning the text appropriately for accurate and efficient character recognition [32]. One of the primary objectives of image processing algorithms is the extraction of valuable information, and a key class of such algorithms focuses on OCR .

This study proposes a voting system that amalgamates the outcomes from Hough Transform, projection profiling, and Frequency Domain Hough Transform into a single consolidated response. To this end, a confidence level has been attached to the output of each algorithm.

Hough Transform Algorithm Steps:

1. Image Preprocessing:
   a. threshold the image
   b. identify contours (letters/clusters of letters) in the thresholded image.
   c. generate a new image placing dots at the same coordinates as the contour anchors identified earlier.
2. Look for lines using the OpenCV [7] V4.12 function HoughLines (the function returns a list of the most voted lines, in descending order of votes).
3. Compute the confidence:
   a. lines receiving fewer votes than a constant divided by the highest vote-getter are removed.
   b. calculate the sum of all votes for the remaining lines.
   c. calculate the sum of the votes for lines with the same angle as the dominant line.
   d. divide the two sums to obtain a number between 0 and 1.



Abstract: Optical Character Recognition (OCR) is an indispensable tool for technology users nowadays, as our natural language is presented through text. We live under the need of having information at hand in every circumstance and at the same time having machines understand visual content and thus enable the user to be able to search through large quantities of text.

To detect textual information and page layout in an image page, the latter must be properly oriented. This is the problem of the so-called document deskew, i.e. finding the skew angle and rotating by its opposite. This paper presents an original approach which combines various algorithms that solve the skew detection problem, with the purpose of always having at least one to compensate for the others' shortcomings, so that any type of input document can be processed with good precision and solid confidence in the output result.

Keywords: deskewing; skew angle detection; automatic document orientation, computer vision, OCR preprocessing; image document analysis

Figure 1.11: From left to right: a. the initial image; b. image with anchor points; c. angle of inclination detection [32].

The steps for the Projection Profiling algorithm are:

1. Pre-process the image:
   a. Threshold the image [11]
   b. Identify contours (letters/groups of letters) in the thresholded image.
   c. Generate a new image by placing points at the same coordinates as the anchors of the contours found in the previous step.
2. Rotate the image and find the angle for which the vertical projection has a maximal variance
3. Compute the confidence
   a. divide the maximal variance by the sum of all variances (which wasfound, empirically, to have values between 0 and 0.5)
   b. subtract 0.5
   c. take the absolute value
   d. multiply by a constant (to normalize the confidence values of the algorithms)
   e. subtract the result from 1

The steps for the Frequency Domain Hough Transform algorithm are:

1. Pre-process the image
   a. pad the image so it becomes square-shaped
2. compute the Fourier Transform
3. create the image of amplitudes
4. threshold the image [12]
5. look for lines using OpenCV HoughLines
6. compute the confidence (the steps are the same as the regular Hough
7. Transform)
8. compute the confidence (the steps are the same as the regular Hough
9. Transform)

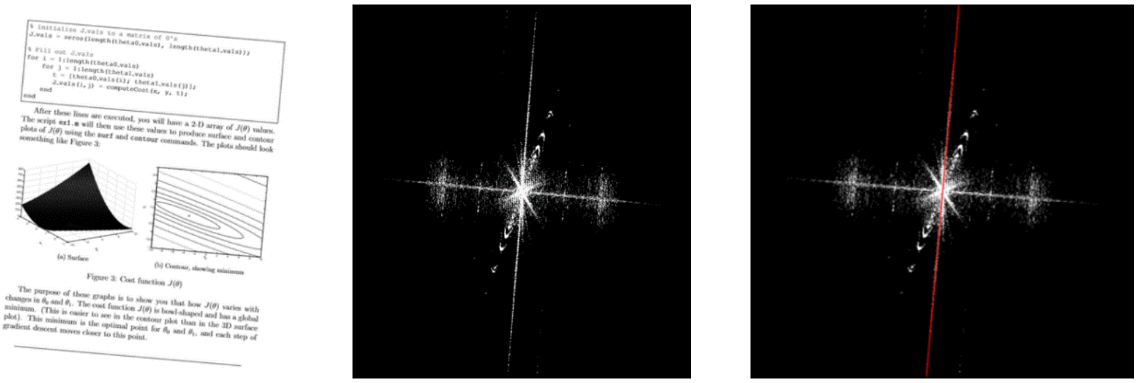


Figure 2.12: From left to right: a. initial image; b. processed FFT; c. skew angle detection [32].

The voting system amalgamates angles and confidences from all these algorithms, delivering an optimal solution. Three voting mechanisms were explored: Best First, Unanimous Vote, and Weighted Vote. The first method selects the algorithm output with the highest confidence, wholly sidelining others. The Unanimous Vote computes the skew angle as the average of each algorithm's outcome. The final method, Weighted Vote, operates similarly but computes a weighted average. Improved outcomes are achievable if results with confidence less than half of the maximum are disregarded.

The "Best First" approach yielded the most favorable outcomes, as can be discerned from Table 2.4.

Deskew Method	**Accuracy**	**Average Error**	**Average Confidence**	**Execution Time (ms)**
Hough Lines	99.150%	0.085°	0.632	4.473
Projection Profiling	99.523%	0.048°	0.545	131.019
Frequency Hough Lines	99.363%	0.064°	0.481	51.512
Best First Voting	**99.568%**	**0.043°**	**0.752**	**187.004**
Unanimous Voting	99.541%	0.046°	0.616	187.004
Weighted Voting	99.553%	0.045°	0.651	187.004

Table 1.2: Compared performance indicators for the presented method against candidates [32].

The results of the voting systems are slightly better than the individual results of the voters. On a dataset of 1414 images, the accuracy went up by 0.04%, which might seem like a very small increase. However, the starting accuracy was already above 99% so the increase in accuracy is significant.

2.5 Assessment and Interpretation of Contrast

Contrast plays an indispensable role in the realm of digital imaging and computer vision, significantly influencing the visual quality and subsequent interpretation of images. The evaluation and quantification of this contrast have become critically imperative across a broad spectrum of applications including medical imaging, remote sensing, and digital photography. This chapter offers a comprehensive analysis of the various metrics available for assessing image contrast, focusing on their underlying formulas, interpretations, advantages, disadvantages, and relevant use-cases. Metrics from histogram-based perspectives, spatial frequency, and statistical standpoints are compared. The computational complexity, precision, and robustness of these metrics are explored, including their sensitivity to noise and other image degradations [33].

Given the pivotal role of contrast, the capacity to accurately measure and quantify it holds paramount significance. This imperative is not only valid for image enhancement techniques but also for the evaluation of image quality and the performance of image processing algorithms [34] [35]. Precise contrast metrics serve as benchmarks for gauging the efficacy of image enhancement algorithms and can guide the development of novel methodologies [36] [37].

n light of the importance of precise contrast assessments, this chapter provides an in-depth analysis of several widely-used contrast metrics. These include both global and local histogram-based metrics [38], spatial frequency-based metrics such as Fourier Transform Contrast (FTC) and Wavelet Transform Contrast (WTC), and statistical metrics such as Root Mean Square contrast, Michelson contrast, and Weber contrast [39] [40].

Each of these metrics has unique characteristics, advantages, and disadvantages that make them suitable for specific applications and scenarios [41].

Histogram-based Metrics:

1. Global Histogram Contrast (GHC):

$$GHC = Max(I) - Min(I) \tag{22}$$

Where $Max(I)$ and $Min(I)$ represent the maximum and minimum pixel intensities in image I [38].

2. Local Histogram Contrast (LHC):

$$LHC(x,y) = Max\big(I(x,y)\big) - Min(I(x,y)) \tag{23}$$

Where $Max\big(I(x,y)\big)$ and $Min(I(x,y))$ represent the maximum and minimum intensity levels of the pixels in image block located at coordinates (x, y).

Spatial Frequency-based Contrast Metrics

1. Fourier Transform-based Contrast (FTC):

$$FTC = \sum |FT\big(I(u,v)\big)|^2 \tag{24}$$

where $FT\big(I(u,v)\big)$ is the Fourier Transform of image I at frequency coordinates (u,v).

2. Wavelet Transform-based Contrast (WTC):

$$WTC = \sum |WT(\mathrm{I(x,y)})| \tag{25}$$

where $WT(\mathrm{I(x,y)})$ is the Wavelet Transform of image I at spatial coordinates (x, y).

Statistical Contrast Metrics

1. Root Mean Square (RMS) Contrast:

$$RMS\ Contrast = \sqrt{\frac{1}{M \cdot N} \sum (I(x,y) - \mu)^2} \tag{26}$$

where I(x,y) indicates the intensity of the pixel at location (x, y) in the image, μ is the mean intensity of the image, and M and N are the dimensions of the image [39].

2. Michelson's Contrast:

$$MC = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \tag{27}$$

where I_{max} stands out for the image's greatest intensity, and I_{min} stands out for the minimum intensity.

3. Weber's Contrast

$$WC = \frac{I - I_b}{I_b} \tag{28}$$

where I represent the intensity of the object, and unde I_b represents the background intensity [37].

Comparison of Contrast Metrics

Contrast metrics have been extensively used in image processing, computer vision, and related fields to evaluate and enhance the quality of images. However, the effectiveness of each metric can vary based on several factors including computational complexity, accuracy, robustness, and sensitivity to noise and other image degradations. In this part, we compare the contrast measures mentioned in this work across different dimensions.

When it comes to computational complexity, global metrics such as Global Histogram Contrast, RMS Contrast, Michelson’s Contrast, and Weber's Contrast typically have a lower computational load. They either require basic statistical calculations or simple operations on pixel intensities[39,42,43].

On the other hand, spatial frequency-based metrics such as Fourier Transform-based Contrast (FTC) and Wavelet Transform-based Contrast (WTC) are more computationally demanding. They require performing either a Fourier Transform or a Wavelet Transform on the image, operations which can be computationally intensive, especially for larger images [44,45].

The accuracy and robustness of a contrast metric depend largely on the type and complexity of the images being evaluated. Global metrics can provide accurate contrast estimates for simpler images, but can overlook local contrast variations in more complex images [34].

Conversely, FTC and WTC can provide more accurate and robust contrast estimates for complex images as they capture local variations in contrast. However, the choice of the Fourier or Wavelet function and other parameters can significantly impact the accuracy of these metrics [35,45].

Regard to sensitivity to noise and other image degradations, histogram-based and statistical metrics tend to be sensitive to extreme values and noise, which can artificially inflate the contrast estimate [36].

Spatial frequency-based metrics such as FTC and WTC are generally less sensitive to noise, as the Fourier and Wavelet Transforms inherently suppress noise within higher frequency bands. However, they can be sensitive to other image degradations such as blurring, which can affect the image's high-frequency components[46].

The choice of contrast metric depends on the specific requirements of the application, including the computational resources available, the complexity of the images, and the level of noise and other degradations present in the images. While no single metric is universally superior, a combination of metrics can often provide a comprehensive and accurate evaluation of image contrast[45].

Image contrast, as a vital aspect of image quality, carries significant implications for image analysis and processing, as well as computer vision. Through this paper, we have presented an in-depth review of a range of contrast metrics, each offering unique advantages and facing particular limitations.

3 Hierarchical Clustering Methodology for Analysis of Page Structure

This research introduces a robust and reliable technique for structuring document image pages hierarchically, harnessing the power of Delaunay triangulation. Central to our approach is the formation of a cluster tree, which encapsulates the page's content through strategically exploiting layout elements arrangements and their relative distances. By applying our technique, we proficiently categorize the page into distinct clusters encompassing images, titles, and paragraphs. The consequent hierarchical framework, founded on the cluster tree, establishes a durable and trustworthy blueprint of the document layout, thereby accelerating document comprehension and examination [47].

The present chapter builds upon the foundational work outlined in [48], incorporating new explanations, enhanced testing scenarios, a more suitable dataset, and an additional post-processing phase designed to further refine the results of the proposed methodology.

The initial step involves data preprocessing to meet the algorithmic requirements. A critical element of this strategy is the utilization of black-and-white documents. Consequently, regardless of the original color scheme, each document is converted to a black-and-white format to conform with these prerequisites [37] [49].

Contour and Segment Generation

Upon input selection, the procedure entails the generation of image segments, termed "connected components" within the image framework. A cluster of interconnected black pixels constitutes a connected component, a determination that can be achieved through a straightforward algorithm (figure 3.1). Starting from an individual black pixel, the algorithm traverses all connected black pixels until only adjacent white pixels remain [48].

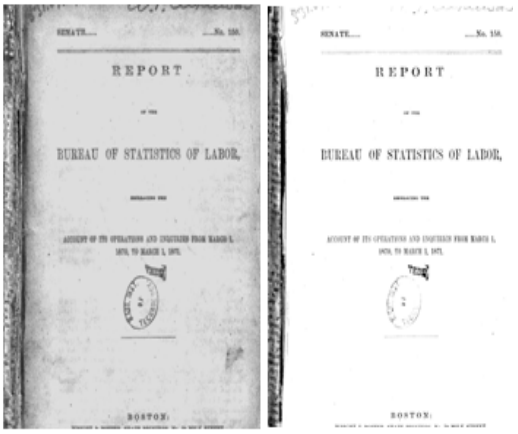
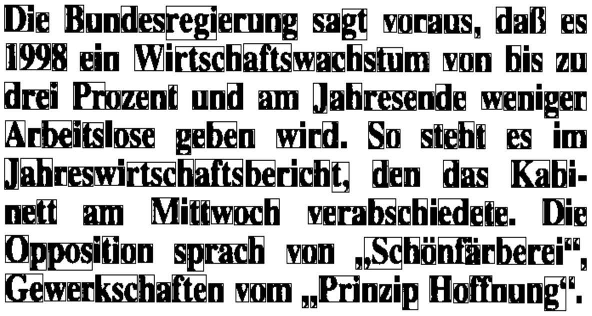


Figure 1.13: Conversion of the initial grayscale image to black and white format [47].

The procedure is iteratively applied to all unvisited black pixels, culminating in the acquisition of connected components. As illustrated in Figure 3.2 and 3.3, a collection of black pixels may be enclosed in various configurations, with rectangles being the most common.



Die Bundesregierung sagt voraus, daß es 1998 ein Wirtschaftswachstum von bis zu drei Prozent und am Jahresende weniger Arbeitslose geben wird. So steht es im Jahreswirtschaftsbericht, den das Kabinett am Mittwoch verabschiedete. Die Opposition sprach von „Schönfärberei“, Gewerkschaften vom „Prinzip Hoffnung“.

Figure 1.14: Bounding rectangles of connected components [47].

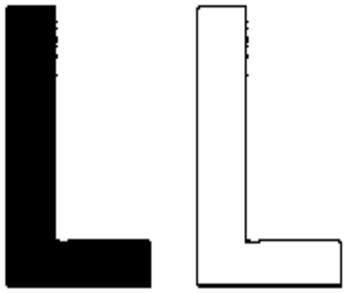


Figure 1.15: Outcome of the contour detection algorithm [47].

Delaunay Triangulation

Upon contour selection, constrained Delaunay triangulation is employed to link all connected components. This results in an excessive number of connections. Hence, a preprocessing phase is executed to eliminate triangles connecting more than two connected components, retaining only paired connected components.

Proximity refers to the relationship between two or more connected components. Through iterative processing of the triangles in the constrained Delaunay triangulation and the exclusion of triangles that connect two distinct connected components (internal triangles), proximity is ascertained. Triangles that are generated within a single connected component (internal triangles) or involve three distinct connected components are eliminated from the analysis.

The Concept of Clustering Tree

Our methodology involves constructing a hierarchical model using a clustering tree, a specialized form of a multi-branch tree. In this tree architecture, connected components serve as leaf nodes, while internal nodes represent clusters of connected components. The diameter of the cluster signifies the maximum distance between any two interconnected components within the same cluster or between adjacent connected components that can form a sequence to connect any two such components (see Figure 3.4). The primary objective of this tree is to categorize connected components into clusters, characterized by increasing diameters. The root of the tree symbolizes the cluster with the largest diameter.

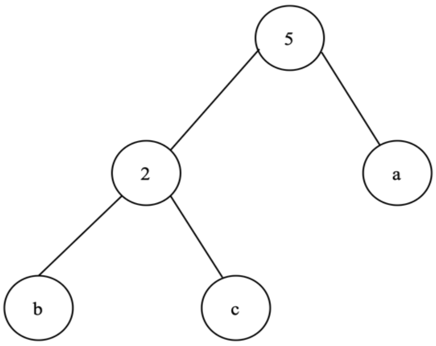
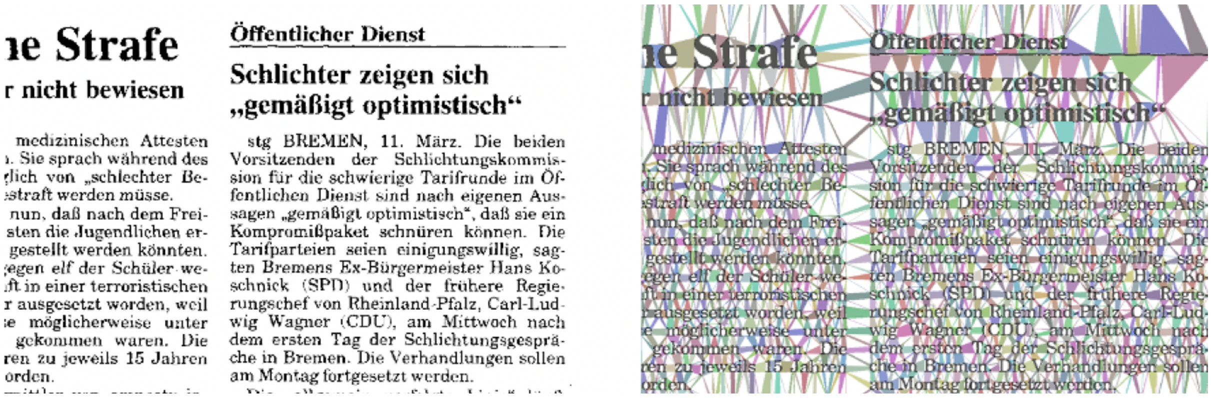


Figure 1.16: A simplified clustering tree where internal nodes are labeled with the diameters of the clusters [47].

The algorithm for constructing the tree commences by calculating the minimum length of the Delaunay triangle's side that connects each pair of connected components. The tree is constructed bottom-up. A cluster is initially created around a random connected component, and the nearest connected component to the initial one is identified and incorporated into the cluster. Subsequently, the nearest connected component to one of the two existing connected components is identified and incorporated into the cluster. If the distance to this newly added connected component is comparable to the distance between the first two, the third connected component is included in the cluster. This procedure repeats until the nearest connected component has such a significant magnitude that it cannot be included in the initial cluster. The remaining clusters are generated using a similar approach, although the algorithm may introduce the closest cluster instead of the closest connected component. The algorithm's outcome is the sought-after tree structure, accurately reflecting the page's hierarchy.



ıe Strafe

r nicht bewiesen

medizinischen Attesten ı. Sie sprach während des ·lich von „schlechter Be- ·straft werden müsse.

nun, daß nach dem Frei- sten die Jugendlichen er- gestellt werden könnten. ;egen elf der Schüler we- ıft in einer terroristischen r ausgesetzt worden, weil ·e möglicherweise unter gekommen waren. Die ren zu jeweils 15 Jahren orden.

Öffentlicher Dienst

Schlichter zeigen sich „gemäßigt optimistisch“

stg BREMEN, 11. März. Die beiden Vorsitzenden der Schlichtungskommission für die schwierige Tarifrunde im Öffentlichen Dienst sind nach eigenen Aussagen „gemäßigt optimistisch“, daß sie ein Kompromißpaket schnüren können. Die Tarifparteien seien einigungswillig, sagten Bremens Ex-Bürgermeister Hans Koschnick (SPD) und der frühere Regierungschef von Rheinland-Pfalz, Carl-Ludwig Wagner (CDU), am Mittwoch nach dem ersten Tag der Schlichtungsgespräche in Bremen. Die Verhandlungen sollen am Montag fortgesetzt werden.

Figure 1.17: The initial image and the result of Delaunay triangulation [47].

A second approach for constructing the hierarchy involves using coordinates derived from the bounding shapes of the connected components and employing the distance between these bounding shapes as a metric. Convex hull is a suitable option for the bounding shape. In this case, the convex hull of each connected component is calculated using its contour points, and the minimum distance between the bounding structures is determined and used, as previously done, as the minimum distance between the connected components. Similar to the earlier method, the algorithm starts with an empty set and develops a cluster by adding connected components with the shortest minimum distance between the bounding structures. Clusters are similarly constructed, with the possibility of including other clusters if their distances are of the same magnitude.

Figure 3.5 illustrates two versions of the same image: one before Delaunay triangulation and the other after its application to provide a visual representation and facilitate understanding. In these images, connected components are visibly linked through numerous edges, accentuating their interconnection. Each color represents the connections between the points on the edge of two connected components, providing a distinct indication of their respective distances. As input for the algorithm, only the connection with the shortest distance is selected from this collection.

The analytical method presented in this document represents a natural evolution of hierarchical clustering procedures. It simulates the viewer's progressive withdrawal from a document, resulting in an image that becomes increasingly "blurry." Although precise details are lost, the document's overall structure remains visible, including the positioning of paragraphs, headings, tables, and images.

Utilizing Delaunay triangulation structures ensures precision is maintained across multiple levels of sampling, faithfully capturing the clustering phenomenon perceived by the human eye as the viewing distance increases.

Furthermore, the human visual system exhibits greater sensitivity to structures characterized by a rectangular shape. Priority is given to the rectangular reconstruction of the clusters in the document by using area functions of the clusters that encourage local maximization.

Following the image analysis of the document, which generated a preliminary structure, a vital post-processing stage is necessary to refine the result, effectively eliminating potentially overlapping elements and ensuring the best representation of structural elements. This process employs Non-Maximum Suppression (NMS), a technique commonly used in artificial vision tasks to reduce the number of overlapping bounding rectangles and retain only the most suitable ones [51] [52].

4 Prediction of Physical Interactions Using Neural Networks

The primary objective is to design, develop, and evaluate an artificial neural network architecture capable of emulating and predicting the dynamic interaction patterns manifested during the encounter between two distinct entities. This chapter principally focuses on the computational learning and understanding of the associated physical impulses that arise when these objects come into contact, elucidating the complex physical interactions involved. Strategically employing an existing physical engine to generate the required training datasets is an integral part of this process, providing a comprehensive and robust foundation for subsequent training and performance evaluation of the neural network [52].

To examine and validate the effectiveness of the proposed artificial neural network model, this study includes a rigorous comparative analysis. The primary focus of this comparison is to juxtapose the results obtained from the trained neural network against those produced by the original physical engine. The aim here is to assess the accuracy, reliability, and practical applicability of the trained model in precisely predicting physical impulses, thereby demonstrating its potential to serve as a viable alternative to traditional physical engines.

A crucial aspect of predicting physical interactions is the appropriate selection and preparation of the dataset used for training the neural networks. This dataset must contain relevant information about the physical interactions between objects or particles, enabling neural networks to learn and make precise predictions. The data preparation process involves: collection of experimental data, data cleaning, data labeling, and dataset splitting.

In this section, different architectures and models of neural networks employed for predicting physical interactions are presented.

Examples of Neural Network Architectures Used in Predicting Physical Interactions:

- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)
- Attention-based Neural Networks

Experimental Application Architecture

To efficiently train the neural network, it is vital to provide input and output data that accurately reflect the logic used by the original physical engine in determining the outcome of a collision between two objects and their subsequent velocities post-impact.

For this purpose, data were collected from the operational physical engine during the execution of various demonstrative tests or "scenarios." These scenarios involved objects undergoing random collisions, and the positional data of the two colliding objects, as well as their velocities and angular velocities at the point of impact, were collected as input data. Correspondingly, the output data consisted of the resulting velocity and angular velocity of the objects following their contact.

The Box2D physical engine [53] was the key tool in the data generation process. In a multi-object collision scenario, the engine conveniently separates the scene into pairs of colliding objects and updates the resulting velocities for each pair individually. This approach ensures that the neural network receives high-quality and representative data, bolstering its ability to accurately predict collision outcomes.

Neural Network Optimization

To achieve a reduced loss function, meticulous refinement of the training dataset was necessary due to its profound influence on the neural network's accuracy. Initially, a randomly generated dataset was used, later transitioning to a well-organized, defined dataset for improved efficiency. In the initial iteration (x, y) point sets were randomly generated within the range [15, 15] for the x-axis. The y-coordinate was consistently set to 15, while angular velocity and rotation were also randomly obtained.

Although this method produced loss functions close to our target, there were certain situations during the neural network's realistic simulation in the engine (where the neural network replaced the traditional logic used for calculating derived velocity at contact) that did not perform as expected. The primary cause of this inconsistent behavior can be attributed to the random nature of dataset generation, which may have led to over-representation of certain scenarios and under-representation of others.

Recognizing this pattern, we decided to abandon randomly generated datasets in favor of an iterative method ensuring equal coverage of a predefined set of cases. This was accomplished by iterating through the initial x-axis range of [-15, 15] with a granular step of 0,05. To aid the network, this range was narrowed down to [-5, 5], including two distinct scenarios: one with a single box hit by a bomb box, and the other involving a stack of 10 boxes subject to the same bomb box impact (the latter implying that the bomb was launched from the same position for both scenarios).

This research delineates an enhanced adaptation of the study presented in [54], executed through meticulous refinements across multiple dimensions.

Firstly, the structure of the training and evaluation dataset underwent significant modifications to better align with computational requirements. This was achieved through a complex blend of data, emphasizing the enrichment of the combination between randomly generated and synthetic datasets. The synthetic data used in this research more closely simulate real data, generating a more robust and versatile dataset. This mixture fosters a more comprehensive learning environment for the neural network, allowing it to effectively extrapolate across a wider range of scenarios and enhance its generalization capabilities.

Secondly, there was a significant shift toward more exhaustive exploration in hyperparameter space, which inevitably led to performance improvements in the neural network. Subtle calibration of hyperparameters is crucial for optimizing the learning process, decisively influencing the overall performance of the neural network model. Therefore, meticulous search and adjustment of these parameters were instrumental in identifying the optimal set that confers superior learning dynamics.

After implementing these constraints, generating training data in accordance with these constraints, and training the neural network with the refined dataset, loss functions in the range of 10-4 were achieved after 5000 training iterations. Table 4.1 illustrates the resulting loss function in a training cycle comprising 2000 iterations.

Iteration	Test Score	Train Score
0	0.105798	0.106044
10	0.051244	0.051222
50	0.015246	0.015388
100	0.007767	0.007789
1000	0.001212	0.001225
2000	0.000728	0.000730

Table 1.3: Comprehensive Results Depicting the Outcome of the Loss Function, Following a Training Cycle Comprising 2000 Iterations, with the Imposed Constraints for Training Data Effectively in Place [52].

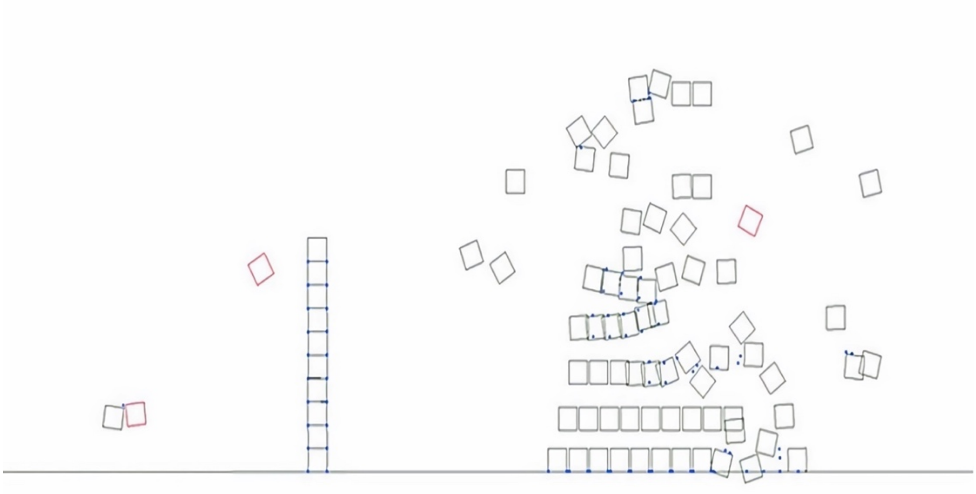


Figure 1.18: The Illustrated Scenarios Used for Training Data Generation; Scenario 1 Demonstrating a Single Object Interaction, Scenario 2 Demonstrating Interaction of a Single Object with a Stack of Ten Additional Objects, and Scenario 3 Highlighting a Complex Interaction Involving a Pyramid Comprised of Sixty-Six Objects [52].

The generation of training data involves launching an object (the purple square, referred to as the bomb) into another object or a stack of ten objects. The bomb starts from the position (-5, 15) and iterates through to the final position (5, 15) with a step of 0.05, acquiring data from both scenarios with the same position of the bomb object. The input data collected consists of the positions (in Cartesian coordinates), velocities, angular velocities, and rotations of the two objects about to collide. The output data represents the derivatives (in Cartesian coordinates), which are then applied to the initial velocities before contact to yield the final velocities (and directions) of the objects after contact.

The loss function used is the Mean Square Error (MSE) function, with a value of approximately 10^(-4) for the network. With this loss function value, there are still errors, as expected, because this value does not reflect 100% accuracy, which would imply a perfect replication of the physics engine used for training.

The network also has a performance impact on the physics engine. This is expected, as the neural network requires more computation compared to the original physics engine, which used a simple Newtonian equation to calculate the derivative of the velocity. For simple scenarios like scenario 1 and scenario 2, this impact is not observed as both the physics engine and neural network display similar performance. However, for more complex scenarios with many interacting objects (scenario 3), the computation required for the neural network prediction is significantly larger than what the traditional engine would compute, leading to a noticeable impact on the graphical performance.

Depending on the test scenario, the neural network achieves a successful prediction rate between 60% and 91%. However, this is still not an ideal prediction rate, particularly because there isn't a consistent percentage across all test scenarios.

Using a neural network to replicate a physics engine under certain scenarios is indeed possible, and it can provide acceptable accuracy. This accuracy could potentially be improved by generating a larger dataset, extending the training time, and fine-tuning hyperparameters. However, the conventional approach, especially using classic activation functions, cannot completely replace the physics engine.

5 Efficient Investments in Software Development

In this chapter, we will delve into the intricacies of software development, an essential and dynamic field that plays a central role in the era of technology and information. Software development involves the complex process of creating, implementing, and managing software applications, which are used across a variety of domains and industries.

The first study is a comprehensive comparison between AAA and Indie computer game production. Although both approaches aim to offer the best possible experience to their audience, the quality of the product can vary depending on various factors, including budget and technology used. To succeed, game developers need to have a clear understanding of the goals that can be accomplished within a reasonable time frame. This study examines the potential profitability of AAA and Indie game development methods. While there is no one-size-fits-all approach to success, we posit that AAA game development generally yields higher profits than Indie game development. Our analysis compares selected game titles in terms of development budget, marketing efforts, team size, technology, and franchise opportunities. Our findings suggest that despite the lower cost of Indie game development, AAA games offer a more secure method of revenue generation [55].

The second study offers a comprehensive comparison between open-source and closed-source projects, examining several key aspects, including the number of contributors or employees, the number of features introduced in the project, the number of vulnerabilities present in the software, revenue or profits, and project management techniques. By comparing these aspects across a series of open-source and closed-source projects, we aim to evaluate the potential for these distribution models to complement each other and to identify the contexts where one model may be more efficient than the other. This analysis aims to provide valuable insights for stakeholders involved in software development, including developers, project managers, and decision-makers [56].

5.1 Game Analysis: Indie vs. AAA

In this subsection, we offer a perspective on the characteristics and differences between "Triple-A" (AAA) and independent (Indie) video game development. The first category comprises large studios and publishers that invest substantial financial resources in creating and promoting high-tech franchises for home gaming consoles. In contrast, Indie development typically involves small teams of up to 15 developers prioritizing creative independence and financial autonomy [57].

AAA Game Ecosystem

It seems that the primary incentive for game developers to seek employment in AAA game companies is the prospect of participating in a large-scale team effort and making a significant contribution to game development. In the context of employment in AAA game companies, financial stability is often cited as a key benefit for employees. Working for a large studio may offer benefits such as medical coverage and flexible scheduling [58]. In AAA game development, a large budget and team do not guarantee a successful game development. The pressure to deliver results for a large team and player base can be significant, requiring individuals to understand the development plan and schedule before committing to a project [59].

Traditional AAA Model

The AAA game development model operates similarly to other business models, with the primary focus on generating profits. This emphasis on profit stems from the considerable financial investments required for the development, testing, and promotion of a single project, given the large scale of these games.

Evolution and Impact of Indie Games

In contrast to AAA games, which require millions of dollars, Indie games are typically developed with much smaller budgets, generally in the range of thousands of dollars. As a result, most Indie game developers operate under resource constraints, often fulfilling multiple roles such as tester, programmer, and designer [60].

Regarding Indie game development, securing sufficient funding can be a challenging task. However, there are various strategies that developers can adopt to address this issue, including: self-funding, crowdfunding, publisher funding, graphic technology, marketing promotion, franchising, and team size.

In the first weeks of January 2023, a survey was conducted to determine people's preferences regarding video games. The survey had 115 participants, consisting of 34 students (30%), 48 people aged between 11 and 18 (48%), and 33 employed adults aged between 30 and 60 (22%).

The survey included the following questions:

1. How many hours per week do you spend playing Indie games?
2. How many hours per week do you spend playing AAA games?
3. Which type of game (Indie or AAA) do you prefer to play and why?
4. On a scale of 0 to 5, how important are graphics and visual effects to you when playing a game?
5. On a scale of 0 to 5, how important are the story and characters to you when playing a game?
6. On a scale of 0 to 5, how important is the multiplayer component of a game to you?
7. Have you ever played an Indie game that you think should have been as popular as an AAA game?
8. Are you likely to recommend an Indie or AAA game to a friend?

The findings of our study indicate that the average number of hours per week spent by students playing Indie video games is approximately 12, while high school students tend to spend slightly fewer hours, around 10, playing such games [55].

The second question of the survey produced remarkable results, differing from those of the previous query. The average weekly hours students spend playing AAA video games dropped to about 10 hours, while people aged between 11 and 18 spent three times more time, averaging 30 hours per week.

Question 3 aimed to investigate the definitive preference of the three groups of participants, and the results align with the findings of questions 1 and 2. The student group displayed a greater inclination toward Indie games, as evidenced by the number of respondents preferring this category over AAA games (24 versus 10). In contrast, high school students preferred AAA games to a greater extent than Indie games (30 versus 18). The adult group, on the other hand, showed a relatively balanced preference, with 19 respondents indicating a preference for AAA games and 13 for Indie games. Regarding the open-ended part of the question, the most common reason for the popularity of Indie games was their distinct artistic style, while for AAA games, the main factor was the ability to engage in multiple player-versus-player (PvP) confrontations.

The next question yielded a conclusive result, as about 40 respondents indicated that graphics and visual effects are significant factors in their gaming experience, regardless of the type of game. The remaining 75 respondents predominantly provided moderate ratings between 2 and 3 on the importance scale, with 45 answers in this range.

Question 5 produced more uniformly distributed answers across the rating scale, with a slight skew toward the lower end of the scale compared to question 4. Specifically, 35 respondents indicated that the story and characters in a game are of high importance (rating of 4 or 5 on the scale), while 28 respondents expressed uncertainty or a lower level of importance (rating of 2 or 3 on the scale).

Question 6 collected mixed responses from the participants, similar to the previous question. Of the 115 respondents, 30 indicated a strong preference for the multiplayer component of video games, while 25 showed little interest in it. The remaining 60 respondents fall into an intermediate zone, suggesting that the importance of multiplayer functionality is highly subjective and depends on the individual player.

Question 7 yielded intriguing results, with popular responses already featuring prominent Indie games like "Stardew Valley" and "Hollow Knight". Respondents wish that Indie games would gain more recognition in the AAA gaming world, with titles like "Celeste" competing with Ubisoft or Rockstar games.

Question 8 is an essential query, given that personal recommendations from acquaintances are influential in shaping game preferences. The survey data highlight the fact that 72 respondents would recommend an AAA game, while 43 would recommend an Indie game.

5.2 Open vs. Closed Aspects in Software Development

The primary objective of this subsection is to carry out an exhaustive comparison between open-source and closed-source software projects. This comparison is based on several key aspects, including the number of contributors/employees, the number of features introduced in the project, the number of vulnerabilities (bugs or more severe issues) present in the software, revenues, and project management techniques. An important metric to consider is the number of contributors, which is relevant due to the fundamental differences in the underlying development models of the two types of projects. Open-source projects rely on contributions from a broad and diverse group of developers, who may participate part-time or as paid employees of other organizations. In contrast, closed-source projects depend on a fixed number of full-time employees who are assigned specific tasks.

Feature analysis becomes significant in evaluating the success of a software project, which directly influences its acceptance by users or investors. The number of features and the development time are critical indicators that determine the speed and efficiency with which the software evolves and is distributed to end-users. For open-source projects, we obtained data from 10 applications, sourced from the TAWOS Dataset [61], which we analyzed and processed according to our research requirements.

Examining multiple open-source projects, we calculated the ratio between contributors and the total number of features for the top three products in a comparable time frame of two years (2018-2020): MongoDB Core Server - 0.37, Moodle - 0.51, Sonatype Nexus - 0.92. Closed-source projects focus on specific features requested by their clients, while open-source projects tend to be more diffuse in their functionality objectives. In the latter, a particular feature may be introduced by a contributor for personal use or for the benefit of a small number of community members, prioritizing practicality over serving a large number of end-users. This distinction can be summarized by the term impact: closed-source projects focus on fewer features with greater impact, while open-source projects expand on a greater number of features with lesser impact.

Security is a crucial aspect of software development and, consequently, it is essential to consider it when writing code. Therefore, our third metric focuses on the security of both closed-source and open-source projects.

Investment Benefits and Risks

Investments in software development are a critical aspect in today's digital economy, as technology plays a pivotal role in business success and innovation. This chapter focuses on the benefits and risks associated with investments in software development, underscoring the importance of a balanced and informed approach to decision-making.

Investment Benefits in Software Development: Improved operational efficiency, increased competitiveness, adaptability and scalability, control over data security, long-term savings.

Investment Risks in Software Development: High costs, risk of failure, development duration, dependency on the development team, need for updating and maintenance.

6 Conclusions and Contributions

Based on the analysis conducted in this doctoral thesis, significant contributions have been made in various technological fields, ranging from the optimization of OCR techniques and motion estimation to software development and page structure analysis. A key aspect of this research was the adoption of an interdisciplinary approach, which enabled the identification of innovative solutions for complex problems. For instance, the weighted voting mechanism improved accuracy in OCR technologies, while contrast analysis and motion estimation opened new avenues for enhancing image processing. Additionally, the study examined the differences between Indie and AAA games, underscoring that innovation and adaptability can often trump substantial financial resources in the development of successful software.

Advancements in the fields studied are not merely theoretical but have direct applicability in various industries. The contrast metrics discussed are vital for domains such as medical imaging, remote sensing, and digital photography. Likewise, understanding and optimizing physical interactions through neural networks can have significant implications in areas such as physical simulations or robotics. However, the work also took on the role of identifying current gaps and limitations, thus providing a framework for future research. For example, although neural networks can enhance physical simulations, they cannot yet fully replace traditional physics engines. This balance between progress made and the identification of areas requiring further improvement makes this thesis not just an academic reference point but also a practical tool for the industry.

6.1 List of Publications

Scientific Articles in Journals and Proceedings of Conferences

ISI Indexed Journals

1. **Cristian Avatavului**, Rares-Cristian Ifrim, Mihai Voncila "CAN NEURAL NETWORKS ENHANCE PHYSICS SIMULATIONS?", BRAIN. Broad Research in Artificial Intelligence and Neuroscience Vol.14 No.2 - 2023, e-ISSN: 2067 – 3957, pp. 76-92 https://doi.org/10.18662/brain/14.2/445
2. **Cristian Avatavului**, Costin-Anton BOIANGIU „A HIERARCHICAL CLUSTER TREE APPROACH LEVERAGING DELAUNAY TRIANGULATION?", BRAIN. Broad Research in Artificial Intelligence and Neuroscience, e-ISSN: 2067 – 3957 [Acceptat în Vol 14(3), 2023, în curs de publicare]

ISI Proceedings of ISI Indexed Conferences

3. Iulia-Cristina Stănica, Costin-Anton Boiangiu, Giorgiana Violeta Vlăsceanu, Marcel Prodan, **Cristian Avatavului**, Răzvan-Adrian Deaconescu, Codrin Tăut, "A Survey on History, Present and Perspectives of Document Image Analysis Systems", in New technologies and redesigning learning spaces Book of abstracts, page 36, 15th eLearning and Software for Education Conference, Bucharest, 2019, ISSN 2360-2198, doi: 10.12753/2066-026X-19-025
4. Giorgiana Violeta Vlăsceanu, Costin-Anton Boiangiu, Răzvan-Adrian Deaconescu, Marcel Prodan, **Cristian Avatavului**, Răzvan Rughiniș, Irina Mocanu, "Designing a Document Image Analysis System on 3 Axis: Education, Research and Performance", in New technologies and redesigning learning spaces Book of abstracts, page 37, 15th eLearning and Software for Education Conference, Bucharest, 2019, ISSN 2360-2198, doi: 10.12753/2066-026X-19-027

Indexed in Other Databases

5. Remus Petrescu, Sergiu Manolache, Costin-Anton Boiangiu, Giorgiana Violeta Vlăsceanu, **Cristian Avatavului**, Marcel Prodan, Ion Bucur; "Combining Tesseract and Asprise results to improve OCR text detection accuracy", The Journal of Information Systems & Operations Management, Vol.13(1), 2019, ISSN 1843-4711, pp. 57-64
6. Alexandru ILINU, **Cristian AVATAVULUI**, Giorgiana Violeta VLĂSCEANU, Costin-Anton BOIANGIU; "VOTING-BASED MOTION ESTIMATION ", The Journal of Information Systems & Operations Management, Vol.14(1), 2020, ISSN: 1843-4711, pp. 82-92
7. Nicolae Tarbă, Daniel Schmidt, Anda - Elena Popovici, Eduard Stăniloiu, **Cristian Avatavului**, Marcel Prodan, “On performing skew detection and correction using multiple experts’ decision”, The Journal of Information Systems & Operations Management, Vol.14 No.2 - 2020 pp. 188-195
8. Robert STANCA, Eduard-Marius COJOCEA, **Cristian AVATAVULUI**, Costin-Anton BOIANGIU "On How To Combine Single Image Super-resolution Algorithms", The Journal of Information Systems & Operations Management, Vol.14 No.1 - 2020, ISSN: 1843-4711, pp. 140-150
9. **Cristian Avatavului**, Marcel Prodan „Evaluating image contrast: a comprehensive review and comparison of metrics”, The Journal of Information Systems & Operations Management, Vol.17 No.1 – 2023, ISSN 1843-4711, pp. 143-160.
10. Giorgiana Violeta Vlăsceanu, **Cristian Avatavului**, Costin-Anton Boiangiu, "An extensive review of metrics for evaluating image binarization", Journal of Information

Systems & Operations Management, Vol 17 No. 1, 2023, ISSN: 1843-4711, pp: 200-222

11. **Cristian Avatavului**, Ilie-Octavian Sandu, Nicu-Cătălin Ioviță, Bogdan Vasile, Ciprian Duță, Costin-Anton Boiangiu, Mihai-Lucian Voncilă and Nicolae Tarbă „Indie vs AAAs: A Fair Comparison", The Journal of Information Systems & Operations Management, ISSN 1843-4711 [Acceptat în Vol 17 (2), 2023, în curs de publicare]
12. **Cristian Avatavului**, Andrei-Iulian Cucu, Alexandru-Mihai Gherghescu, Costin-Anton Boiangiu, Iulia-Cristina Stănica, Cătălin Tudose , Mihai-Lucian Voncilă, Daniel Rosner „Open-source and closed-source projects: A fair comparison", The Journal of Information Systems & Operations Management, ISSN 1843-4711 [Acceptat în Vol 17 (2), 2023, în curs de publicare]

Posters

13. **Cristian AVATAVULUI**, "Combining Tesseract and Asprise results to improve OCR text detection accuracy" – in Semicentennial Anniversary of the Department of Computers, University Politehnica of Bucharest 2018.

Bibliography

1. Mueller JP, Massaron L. Artificial Intelligence For Dummies. 1st ed. For Dummies; 2018.

2. Szeliski R. Computer Vision: Algorithms and Applications. Springer; 2010.

3. Petrescu R, Manolache S, Boiangiu CA, Avatavului C, Prodan M, Bucur I, Vlasceanu G. Combining Tesseract and Asprise results to improve ocr text detection accuracy. Journal of Information Systems Management. 2019 May 1;13:57.

4. Bhowmick A, Govindaraju V. A Framework for Evaluation of Uncertain Inputs in OCR Systems. In: Proceedings of the 8th IAPR International Workshop on Document Analysis Systems. 2008.

5. SDK AOCR. Asprise OCR and its applications. Asprise; 2020.

6. Wshah S, Kumar G, Govindaraju V. Script Independent Word Spotting in Offline Handwritten Documents. In: International Conference on Document Analysis and Recognition. 2011.

7. Graves A, Schmidhuber J. Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks". In: Advances in Neural Information Processing Systems 21. 2009.

8. GitHub - tesseract-ocr/tessdoc: Tesseract documentation [Internet]. [cited 2023 Feb 1]. Available from: https://github.com/tesseract-ocr/tessdoc

9. In: OCR Community Help Wiki, URL [Internet]. Available from: https://help.ubuntu.com/community/OCR,

10. Graves A, Jaitly N. Towards End-to-End Speech Recognition with Recurrent Neural Networks". In: Proceedings of the 31st International Conference on Machine Learning. 2014.

11. Park SC, Park MK, Kang MG. Super-Resolution Image Reconstruction: A Technical Overview". IEEE Signal Processing Magazine. 2003;

12. Canny J. A Computational Approach to Edge Detection. IEEE Trans Pattern Analysis and Machine Intelligence. 1986;8(6):679–98.

13. Ilinu A, Avatavului CD, Vlăsceanu GV, Boiangiu CA. Voting-Based Motion Estimation. Journal of Information Systems & Operations Management. 2020;14.

14. Moeslund T, Granum E. A survey of computer vision-based human motion capture. Computer Vision and Image Understanding. 2001 Mar 1;81:231–68.

15. J.R. B, P. A, K.J. H, R H. Hierarchical modelbased motion estimation. In: Sandini G, editor. Computer Vision. p. 92.

16. Cédras C, Shah M. Motion-based recognition a survey. Image and Vision Computing [Internet]. 1995 Mar 1;13(2):129–55. Available from: https://www.sciencedirect.com/science/article/pii/026288569593154K

17. A. French Optical Flow. Computerphile, editor. 2019.

18. A IE. Bobick Introduction to Computer Vision, udacity.com. 2015.

19. Shah M. Optical Flow. UCF Computer Vision Video Lectures; 2012.

20. Zhu S, Ma KK. A new diamond search algorithm for fast block-matching motion estimation. IEEE Transactions on Image Processing. 2000 Feb;9(2):287–90.

21. Fundamentals of Digital Image and Video Processing [Internet]. Coursera. Available from: https://www.coursera.org/learn/digital

22. de Haan G, Biezen PWAC, Huijgen H, Ojo OA. True-motion estimation with 3-D recursive search block matching. IEEE Transactions on Circuits and Systems for Video Technology. 1993 Oct;3(5):368–79.

23. Min Li, Biswas M, Kumar S, Truong Nguyen. DCT-based phase correlation motion estimation. In: 2004 International Conference on Image Processing, 2004 ICIP '04 [Internet]. Singapore: IEEE; 2004. p. 445–8. Available from: http://ieeexplore.ieee.org/document/1418786/

24. Sun D, Roth S, Black M. Secrets of Optical Flow Estimation and Their Principles. In 2010. p. 2432–9.

25. Torr PHS, Zisserman A. Feature Based Methods for Structure and Motion Estimation. In: Triggs B, Zisserman A, Szeliski R, editors. Vision Algorithms: Theory and Practice. Berlin, Heidelberg: Springer; 2000. p. 278–94. (Lecture Notes in Computer Science).

26. Argyriou V, Vlachos T. A study of sub-pixel motion estimation using phase correlation. In 2006. p. 387–96.

27. Xu L, Jia J, Matsushita Y. Motion Detail Preserving Optical Flow Estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012 Sep;34(9):1744–57.

28. Bruenig M, Niehsen W. Fast full-search block matching. Circuits and Systems for Video Technology, IEEE Transactions on. 2001 Mar 1;11:241–7.

29. Nie Y, Ma KK. Adaptive rood pattern search for fast block-matching motion estimation. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society. 2002 Feb 1;11:1442–9.

30. Freeman WT, Jones TR, Pasztor EC. Example-based super-resolution. IEEE Computer Graphics and Applications. 2002;22(2):56-65,.

31. STANCA R, COJOCEA EM, AVATAVULUI C. Costin-Anton BOIANGIU "On How To Combine Single Image Super-resolution Algorithms." The Journal of Information Systems & Operations Management. 14(1–2020, ISSN):1843–4711, 140–50.

32. Tarbă N, Schmidt D, Popovici AE, Stăniloiu E, Avatavului CD, Prodan M. On Performing Skew Detection and Correction Using Multiple Experts' Decision. Journal of Information Systems & Operations Management. 2020;14.

33. Avatavului CD, Prodan M. Evaluating Image Contrast: A Comprehensive Review and Comparison of Metrics. Journal of Information Systems & Operations Management. 2023;17.

34. Li C, Bovik AC. Three-component weighted structural similarity index. In: Proceedings of SPIE - The International Society for Optical Engineering. 2009.

35. Kovesi P. - Image features from phase congruency - Videre. Journal of Computer Vision Research. 1999;1(3):1–26.

36. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. - Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing. 2004;13(4):600–12.

37. Otsu N. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics. 1979;9(1):62–6.

38. Zuiderveld K. - Contrast Limited Adaptive Histogram Equalization. In: Heckbert P, editor. Graphics Gems IV. Academic Press; 1994. p. 474–85.

39. D. Shaked, Tastl I. Sharpness measure: Towards automatic image enhancement. In: Proceedings of the IEEE International Conference on Image Processing. 2005.

40. Asamoah D, Oppong E, Oppong S, Danso J. Measuring the Performance of Image Contrast Enhancement Technique. International Journal of Computer Applications. 2018;181:6–13.

41. PIELLA G, HEIJMANS HJAM. A new quality metric for image fusion. In: Proceedings of the IEEE International Conference on Image Processing. 2003.

42. Gonzalez RC, Woods RE. - Digital Image Processing. 4th ed. Pearson; 2018.

43. HARALICK RM, SHAPIRO LG. - Computer and Robot Vision - Volume I. Addison-Wesley; 1992.

44. Matheron G. - Random sets and integral geometry - Wiley. 1975.

45. STARK JA. Adaptive image contrast enhancement using generalizations of histogram equalization. 889-896. IEEE Transactions on Image Processing. 2000;9(5).

46. Rahman Z, Jobson DJ, Woodell GA. - Multiscale Retinex for color image enhancement. Image Processing - IEEE Transactions on. 2004;13(7):1005–13.

47. Avatavului CD, Boiangiu CA. A hierarchical cluster tree approach leveraging Delaunay triangulation. BRAIN Broad Research in Artificial Intelligence and Neuroscience. 14(3).

48. Boiangiu CA, Cananau DC, Raducanu B, Bucur I. A Hierarchical Clustering Method Aimed at Document Layout Understanding and Analysis. International Journal of Mathematical Models and Methods in Applied Sciences. 2008 Jan 1;2:413–22.

49. Sauvola J, Pietikäinen M. Adaptive document image binarization. Pattern Recognition [Internet]. 2000 Feb 1;33(2):225–36. Available from: https://www.sciencedirect.com/science/article/pii/S0031320399000552

50. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017 Jun;39(6):1137–49.

51. Malisiewicz T, Gupta A, Efros AA. Ensemble of Exemplar-SVMs for Object Detection and Beyond. In: 2011 International Conference on Computer Vision. 2011.

52. Avatavului CD, Ifrim RC, Voncila M. Can Neural Networks Enhance Physics Simulations? BRAIN Broad Research in Artificial Intelligence and Neuroscience. 2023 Jun 30;14(2):76–92.

53. Catto E. Fast and Simple Physics using Sequential Impulses. In: Proceedings of Game Developer Conference. 2006.

54. Ifrim RC, Penariu P, Boiangiu CA. Replicating Impulse-Based Physics Engine Using Classic Neural Networks. Journal of Information Systems & Operations Management. 2021;15(2).

55. Avatavului CD, Prodan M. Indie vs AAAs: A Fair Comparison. Journal of Information Systems & Operations Management. 2023;18.

56. Avatavului CD. Open-Source and Closed-Source Projects: A Fair Comparison. Journal of Information Systems & Operations Management. 2023;18.

57. Karthikeya K. The Difference Between Working in Indie and AAA Game Development [Internet]. Gameopedia; 2022. Available from: https://www.gameopedia.com/indie-aaa-aa-games-comparison

58. Shoshanah WALL. Should I Work for a AAA or Indie Video Game Studio?, CGSpectrum, 2022 [Internet]. Available from: https://www.cgspectrum.com/blog/difference-between-aaa-vs-indie-game-studio

59. Campbell C. What aaa can learn from indies – according to indies [Internet]. 2023. Available from: https://www.gamedeveloper.com/audio/what-aaa-can-learn-from-indies---according-to-indies

60. Mathews CC. Nia WEARN - How Are Modern Video Games Marketed? The Computer Games Journal. 2016;5:23-37,.

61. Tawosi V, Al-Subaihin A, Moussa R, Sarro F. A versatile dataset of agile open source software projects. In: Proceedings of the 19th International Conference on Mining Software Repositories [Internet]. Pittsburgh Pennsylvania: ACM; 2022. p. 707–11. Available from: https://dl.acm.org/doi/10.1145/3524842.3528029