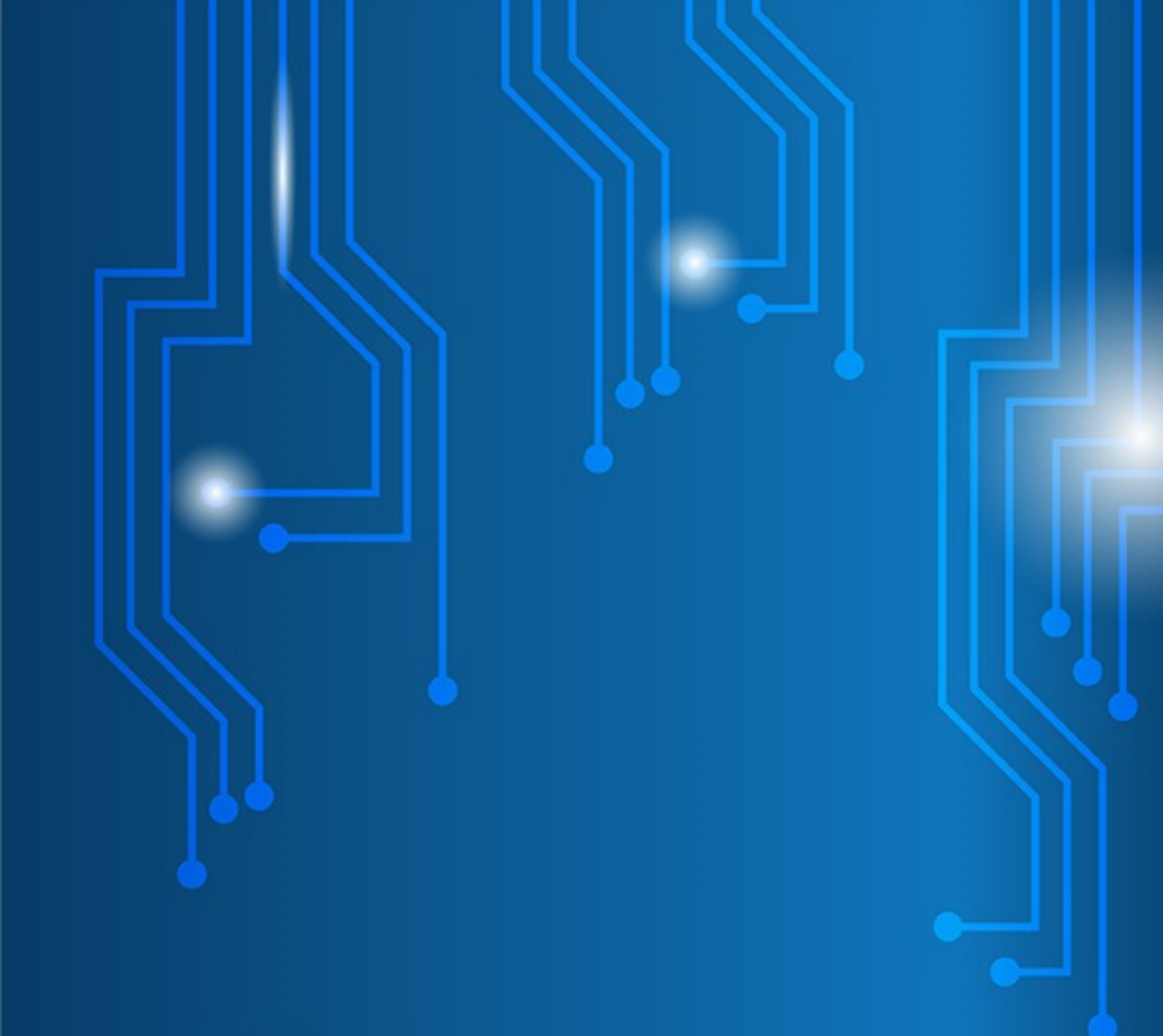




**Journal of University of Information Science
and Technology “St. Paul the Apostle” - Ohrid
ISSN: 2955-182X**

Publisher University of Information Science and Technology “St. Paul the Apostle” - Ohrid

Volume 4 | Issue 1 | 2024

EDITORIAL - Preface to Volume 4 Issue 1 of the Journal of University of Information Science and Technology “St. Paul the Apostle” – Ohrid

Dear Readers,

Journal of University of Information Science and Technology “St. Paul the Apostle” – Ohrid is an international, peer-reviewed journal published annually since 2016. It is an open access Journal available at the website of the University of Information Science and Technology “St. Paul the Apostle” – Ohrid (<https://journal.uist.edu.mk/>)

This Journal is committed to publishing and disseminating high-quality and novel scientific research work in the broad field of Applied Mathematics, Automation & Robotics, Blockchain, Business, Economy, Entrepreneurship & Management, Communication & Network Technology, Cybersecurity, Data Science, Distributed and Parallel Computing, Digital Art and New Media Art, Metaverse, Virtual and Augmented Reality, Communication & Network Technology, Smart Systems & IoT, Information Society & Digital Transformation, Biomedical Engineering.

As editor-in-chief of the Journal of the University of Information Science and Technology “St. Paul the Apostle” – Ohrid, I am pleased to present Volume 4, an issue mainly devoted to the classification, convolutional neural networks, Simulation and animation 3D avatars, scalability and performance for evaluating server capabilities, blockchain technology and virtual non-fungible token gallery. We thank all the authors for contributing to this Issue and all the reviewers for providing detailed and timely evaluations of the submitted manuscripts.

It is an excellent challenge for the editor members to bring a new journal issue into the world, primarily when the Journal aims to publish high-quality manuscripts. I want to express my sincere gratitude to all of them for their excellent work, remarkable contribution, enthusiasm and support.

Sincerely Yours,

Editorial Board

February 2024

Ohrid



Journal of University of Information Science
and Technology "St. Paul the Apostle" - Ohrid
ISSN: 2955-182X
IMPRESUM

FOUNDER AND PUBLISHER

University of Information Science and Technology "St. Paul the Apostle" – Ohrid

EDITOR IN CHIEF

Professor Ninoslav Marina, PhD

EXECUTIVE EDITOR

Associate Professor, Atanas Hristov, PhD

EDITORIAL BOARD

Professor Arban Uka, Epoka University – Tirana, Albania

Professor Ustijana Reckoska Shikoska, PhD, UIST "St. Paul the Apostle" – Ohrid, Macedonia

Professor Ishwar K. Sethi, PhD Oakland University, USA

Professor PREM K. KALRA, PhD Head & Professor, Indian Institute of Technology, Kanpur, India

Professor Eric Châtelet, PhD Université de technologie de Troyes, France

Professor Weiler A. Finamore, PhD Pontifical Catholic University of Rio de Janeiro, Brazil

Professor Katsumi Yamashita, PhD University of Osaka, Japan

Professor Mohammad Reza Asharif, PhD University of the Ryukyus, Okinawa, Japan

Professor Heung Gyoon Ryu, PhD Chungbuk National University, Seoul, South Korea

Professor Abdel-Badeeh M. Salem, PhD, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt

Associate Professor, Dijana Capeska Bogatinoska, PhD UIST "St. Paul the Apostle" – Ohrid, Macedonia

Associate Professor, Aneta Velkoska, PhD UIST "St. Paul the Apostle" – Ohrid, Macedonia

Associate Professor, Jovanka Damoska Sekuloska, PhD UIST "St. Paul the Apostle" – Ohrid, Macedonia

Assistant Professor, Aleksandar Karadimche, PhD UIST "St. Paul the Apostle" – Ohrid, Macedonia

Assistant Professor, Jovan Karamachoski, PhD UIST "St. Paul the Apostle" – Ohrid, Macedonia

Assistant Professor, Ljubinka Sandjakoska, PhD UIST "St. Paul the Apostle" – Ohrid, Macedonia

ISSN 2955-182X

CONTENTS

O. Petan, A. Karadimche, D. Hadley

Single-Label COVID-19 Lung Disease Classification 5

A. Rushiti, R. Kaba, S. Rexhepi

Comparison of threading and non-threading implementations in the Pascal Triangle 22

M. Trpkoski, A. Sefidanoski, S. Mitevski, A. Rushiti

Simulation and animation of a 3D avatar from a realistic human face model 32

D. Runchevski, A. Hristov, A. Rushiti

Mathematical modelling of scalability and performance testing methods for evaluating server capabilities
48

J. Ilovski, J. Karamachoski

Gamification of NFT collections: Virtual NFT gallery 56

Single-Label COVID-19 Lung Disease Classification

Olga Petan[†], Aleksandar Karadimche[‡], Dexter Hadley[★]

Email(s): olga.petan@gmail.com, aleksandar.karadimce@uist.edu.mk, dexter.hadley@ucf.edu

[†] University of Information, Science, and Technology Saint Paul the Apostle, Ohrid, 6000, North Macedonia

[‡] University of Information, Science, and Technology Saint Paul the Apostle, Ohrid, 6000, North Macedonia

[★] College of Medicine, Orlando, Florida, United States

I. Abstract

Chest imaging (X-ray, CT, MRI, etc.) is often used to diagnose COVID-19 infections, and the clinical interpretation of these images may influence patient treatment plans including quarantine, ICU admission, and ventilator support. Convolutional neural networks represent state-of-the-art artificial intelligence (AI) for image classification and require large and diverse training image datasets to achieve clinical grade performance. The artificial intelligence community has been focusing on building CNNs to help physicians images for COVID-19 patients, but the open community attempts have been plagued by a few main problems that are characteristic of training AI for clinical applications:

Our work suggests that the incorporation of finer clinical features such as multi-class labeling of outcomes may improve clinical performance. Much more diverse and properly annotated data, as well as a proper data architecture is needed to build clinically useful AI applications. Leveraging computers and state-of-the-art artificial intelligence could play a crucial role in helping physicians manage the spread of COVID-19, as well as provide better and more timely care for patients infected with COVID-19.

The experiments described in this paper investigate whether using computer vision to detect patterns in pathological lung tissue can alleviate the pressure on healthcare systems and facilitate the management of future pandemics. Due to a relatively small dataset, we use the VGG-16 network to do transfer learning on our images. Our results show that Artificial Intelligence can be used to find pathology in patients and to determine whether there is viral or bacterial pneumonia. However, AI shows challenging results when it comes to determining specific viral or bacterial pneumonias, which is in part due to the lack of available data.

Keywords: single-label classification, convolutional neural networks, covid-19, pneumonia, lung disease, artificial intelligence, Microsoft Azure

II. Introduction

It has been a couple of years since humanity found itself in the midst of a global pandemic. What started in late December 2019 in Wuhan as an isolated incident has now resulted in almost 700 million infections and over 7.95 million deaths¹ worldwide at the time of writing. Professionals across the globe were working on developing effective treatment and vaccine distribution, yet one issue remains – how to stop the spread of a virus and how to manage effects of a pandemic, this one or future ones, on the population, medical workers, essential workers, and hospitals.

The artificial intelligence field tried to help this battle too. Many researchers have written hundreds of papers and created hundreds of deep learning models that aim at diagnosing COVID-19 from images, as well as assessing the severity of the disease for prognostic purposes. However, building AI clinically usable AI solutions has been far from straightforward. First, the lack of data is the most prominent obstacle [1,2]. Second, when clinical data is made available, a lack of structured meta-data or any ontological classification of clinical outcomes, severely limits the learning ability of deep learning algorithms and severely biases their performance [3]. Third, a lack of detailed clinical curation such as the severity of infection precludes the clinical applicability of DL models.

While there have almost 700 million cases globally, the AI community only has access to less than 2000 COVID-19 positive images. At the time of writing, there are half a dozen open source image repositories available to AI researchers. For the experiments in this paper, we combined four such repositories and ended up with 4708 X-Ray images, 1764 of which are images of COVID-19 patients. We decided to only use X-Ray images because CT scans are costly both in terms of money and operating skills, and because poorer regions have a greater number of COVID positive patients and deaths. Although X-Ray images are less sensitive for diagnosis compared to CT scans, 69% compared to over 90% respectively [4], X-Ray machines are more widespread, more radiologists know how to operate these machines, require less time for disinfection after being used, and have lower levels of radiation.

Out of those 4708 X-Ray images, only 485 have information about patient location, and almost 87% of the images come from Europe and Asia. The situation is similar with information about the patient's sex – only about 15% of the images have information about the patient's sex, and there is twice as many male patients' images than females'. There is not enough information about patients' age either, as we only have 429 age records. Out of these, there are no records for patients younger than 19. From this, we can see that most of our images are from European and Asian men over the age of 56, and models created on these poorly annotated images will not generalize well and will not be useful for people in the US. This is troublesome, considering that the US has had over 110 million positive cases to date, and over 1.1 million deaths.

¹ https://www.worldometers.info/coronavirus/?utm_campaign=homeAdvegas1?

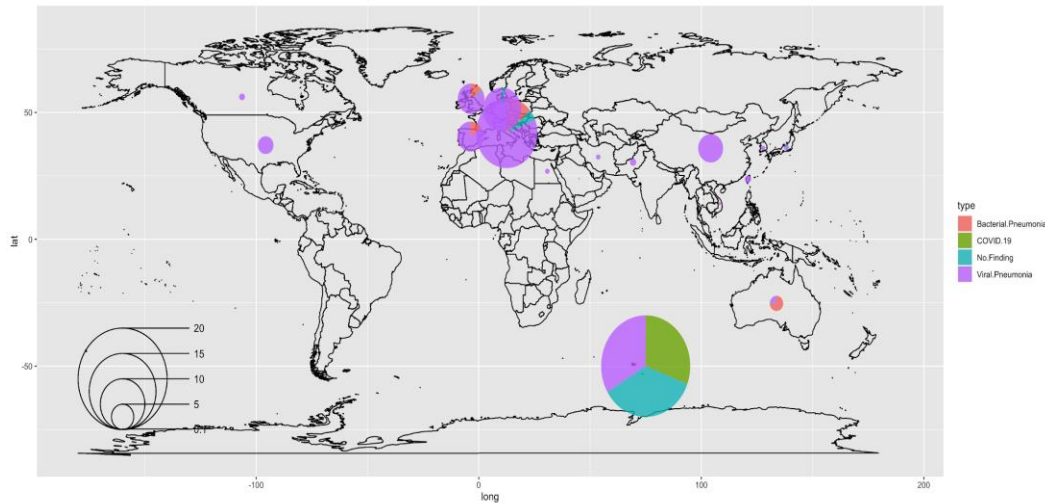


Figure 1: Geographical distribution of labels

We created four experiments to demonstrate the need for much more data. All experiments share the same architecture and are built using the same pre-trained model, but they differ in how the datasets are structured. The first experiment is a binary classification deep learning model that checks whether the X-Ray can pick up pathology on an image in the first place. The second experiment is a multiclass classification model differentiating between COVID-19 positive X-Ray images, bacterial pneumonia, viral pneumonia, and healthy images. The purpose of this experiment is to see whether a deep learning model built on this data can differentiate COVID-19 from other pneumonias. The third experiment is a multiclass classification model trying to differentiate between different pneumonias: bacterial, viral, lipoid, and fungal. The purpose of this experiment is to see whether a deep learning model can find enough differences between different pneumonias. Finally, the third experiment is a multi-class classification model differentiating between different viral pneumonias: influenza, MERS, SARS, COVID-19. The purpose of this experiment is to see whether we can really trust these types of multiclass classification models, and to discuss where the AI community's effort should be focused. All four experiments combined stress the need for acquiring more data and involving physicians in the annotation and etymological processes.

The paper is structured as follows: in Section III – Rationale we discuss why we need more data and why current available data is not useful to build deep learning models. In Section IV – Related Work, we discuss the current deep learning models differentiating between viral pneumonia, bacterial pneumonia, COVID-19, and healthy images. In Section V - Dataset Construction, we discuss how we constructed the dataset we are using in the experiments and describe some metadata about it. In Section VI - Research Methodology, we discuss the four experiments we created and the different models we built for each experiment. And in Section VII – Conclusion, we discuss the results of our experiments and we propose steps which should be taken to build artificial intelligence models to help physicians fight future pandemics.

III. Rationale

If our goal is to develop a clinically usable deep learning model to diagnose COVID positive patients, we need to have a very big input dataset, because neural networks are data-hungry. Even though

there have been about 110 million COVID-19 cases in the US to date, there are less than 2000 available X-Ray images for the AI community to build models with. A substantial problem emerges from this lack of data: the model memorizes instead of learning the disease's features, which will lead to overfitting and the model not being able to classify new data correctly.

To overcome the small number of available X-Ray images problem we use transfer learning, where we repurpose an already trained model and apply it to initialize the weights in our COVID-10 classification model. This, in turn, creates a new problem. ImageNet is the pre-trained model that is most often used in COVID-19 image classification work, and it contains over 14 million images and 1000 classes. However, these are natural images of things like animals, food, landscape, buildings. These domains differ greatly from medical imaging, which is why we need access to a lot more data than we already have.

IV. Related Work

Because chest radiographs are relatively cheaper and require less specialized skills, X-ray imaging is more widely used in COVID-19 imaging. Ground glass opacities, bilateral interstitial opacification, and consolidation are the most frequent manifestation of COVID-19 in an X-ray image [5]. However, this image modality has certain shortcomings that radiologists and physicians must be aware of: images of one patient may vary as the disease develops; and images can be normal even when a test is positive. That is why in severe Covid-19 there is a proportionately greater lung involvement which tends to be denser peripherally and in the lower zones. The role of CXRs as the initial radiological assessment of patients presenting with respiratory distress and possible Covid-19 is established [6].

Feature extraction is also done in X-Ray images. For example, [7] first use the DenseNet103 is used for lung segmentation, and then from the segmented area, Res-Net18 is used to classify diseases using the patches. Because of the small number of positive COVID-19 X-ray images, a transfer learning model approach is taken, which borrows weights from pre-trained deep networks and use those to extract features from images in the small dataset. Thus, [8] use a few pre-trained models to test their limited dataset: VGG16, GoogleNet, ResNet18, InceptionV3, DenseNet. The features extracted from the lung segmentation are fed into an SVM, whose goal is to further classify lung diseases.

V. Dataset Construction

We collected images from four open source repositories: Covid chestxray dataset [9], Figure 1 COVID chestxray dataset [10], Actualmed COVID chestxray dataset [11], and Kaggle's COVID-19 Radiography Database [12]. The distribution of images from these sources is shown in Table 1.

Table 1: Number of images per image source

Data Source	Number of X-Rays
Kaggle	3885
Cohen	633
Actualmed	185
Figure1	5
Grand Total	4708

From these repositories, we collected 4708 images that had 13 lung disease labels and 5 views. The dataset also includes a column indicating whether the patient has had a PCR test, and if they did, whether it was positive or unclear. All images are from adult patients. Figure 2 shows the breakdown of the diseases and how many images were used in the experiments.

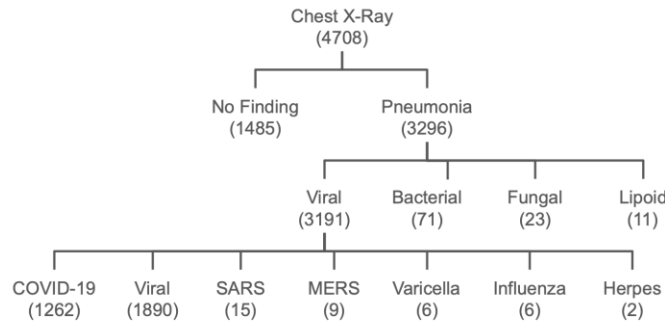


Figure 2: Diseases and number of images per disease

The dataset contains X-ray images from 5 views: PA, AP, AP Supine, Lateral view, and AP Erect view. A short explanation of the different X-Ray views is given below, and Table 4 shows the number of images per view:

- **PA (Posteroanterior) view:** The patient faces the X-ray detector, and the X-ray beam passes through the posterior aspect of the body to the anterior side.
- **AP (Anteroposterior) view:** The patient faces the X-ray source, and the X-ray beam passes from the anterior to the posterior side of the body. Used in situations where the patient cannot be positioned for a PA view, such as trauma cases
- **AP Supine view:** Similar to the AP view, but the patient is lying down (supine) rather than standing. Often used when a patient cannot stand, and a horizontal position is necessary.
- **Lateral view:** The X-ray beam passes from one side of the body to the other, with the patient positioned sideways. These views provide additional information and a different perspective on lung involvement and can help in assessing the extent and distribution of pulmonary abnormalities.
- **AP Erect view:** Similar to the AP view, but the patient is in an upright or standing position.

Table 2: Number of images per view

Views	Number of Images
PA	412
AP	212
AP Supine	146
L	48
AP Erect	1
Grand Total	819

Figure 3 shows some examples of images for each of the views.

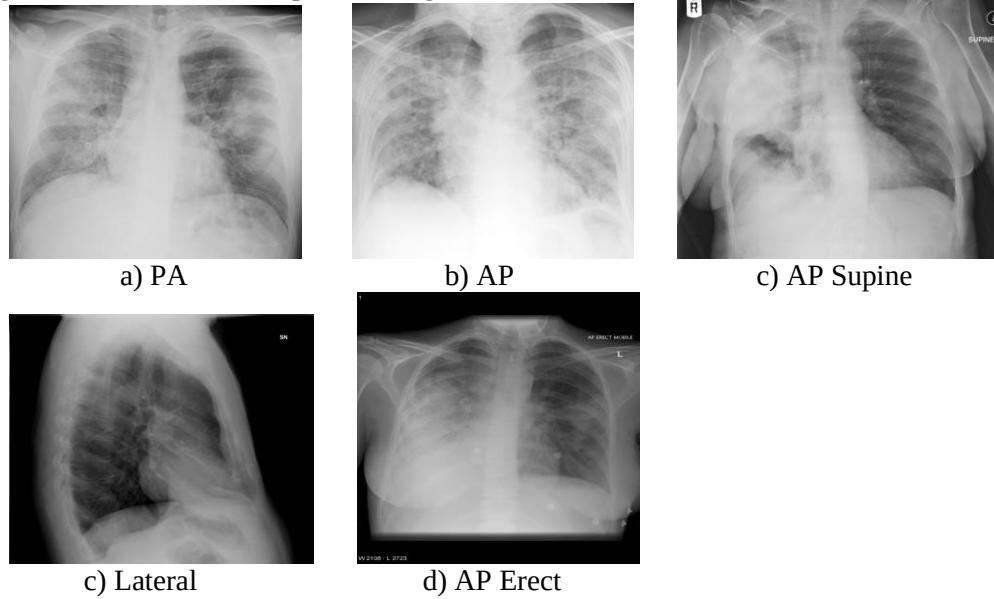


Figure 3: Examples of images per view

We have identified 3 issues with data imbalances in the available image sources: view imbalances where not all diseases have enough views for the model to be trained on; sex imbalances where the metadata for the images is not tagged with the patient's sex or one of the sexes has many more images; geographical imbalances with Europe having the most images and Asia trailing second, but not many images for the rest of the continents; and age imbalances where the average age per continent is over 50. Below we show the distributions of the metadata for images and patients.

The view imbalance issue becomes apparent when we see the image views we have for each disease. For example, we have images for mostly all views for Bacterial and Viral pneumonia, but we only have images for 2 views for COVID-19. Table 3 shows the number of views we have for each disease.

Table 3: Number of images per view and disease

Views	Number of Images	Bacterial Pneumonia	COVID-19	Healthy	Viral Pneumonia
PA	412	35	32	132	213
AP	212	12	27	9	164
AP Supine	146	5	0	3	138
L	48	18	0	0	30
AP Erect	1	1	0	0	0
Grand Total	819	71	59	144	545

The sex imbalance is shown in Table 4. We can see that we do not know the sex of the patients for most of the images. When we do know the sex, the only labels we know it for are Bacterial and Viral pneumonia, and there is a bias towards male patients.

Table 4: Number of images per sex and disease

Sex	Bacterial Pneumonia	COVID-19	Healthy	Viral Pneumonia	Grand Total
NA	1	1262	1469	1401	4133
M	47	0	9	319	375
F	23	0	7	170	200
Grand Total	71	1262	1485	1890	4708

The geographical imbalance is shown in Table 5. We can see that we have very little images tagged with sex or location, and mostly have images from men from Europe and Asia.

Table 5: Number of images per sex and continent

Continent	F	M
Europe	121	237
Asia	46	72
North America	15	33
Oceania	16	14
NA	2	18
South America	0	1
Africa	0	0
Grand Total	200	375

From Table 6, we can further see that we mostly have age information for patients from Europe and Asia and that Europeans on average are slightly older.

Table 6: number of images per continent and information about patients' age per continent

Continent	Average of age	Min of age	Max of age	Number of X-Rays
NA	59.0	28	84	4110
Europe	58.3	20	94	370
Asia	53.2	23	81	137
North America	51.9	21	83	50
Oceania	50.6	25	75	30
Africa	0.0	0	0	10
South America	34.0	34	34	1
Grand Total	56.0	20	94	4708

From Table 7, we can see that most of the metadata we on age and location is for viral pneumonia, and the healthy and COVID-19 images are not tagged.

Table 7: number of images per continent and disease

Continents	Bacterial Pneumonia	COVID-19	Healthy	Viral Pneumonia
NA (avg. age 58.9)	1	1262	1469	1378
Europe (avg. age 58.2)	46	0	11	313
Asia (avg. age 53.2)	1	0	4	132
North America (avg. age 51.8)	0	0	0	50
Oceania (avg. age 50.6)	23	0	1	6
Africa	0	0	0	10
South America (avg. age 34)	0	0	0	1
Grand Total	71	1262	1485	1890

VI.

Research Methodology

In computer vision, some of the most common parameters used when training a model to predict outcomes are:

- **iteration** - the number of times the entire dataset is processed by the model during training;
- **epoch** - a single pass through the dataset during training. During each epoch, the weights of the model are adjusted, which helps with the model accuracy and performance;
- **batch size** - the number of images that the model sees in one iteration. Since the number is usually small, the weights for the model can be updates faster and more frequently;
- **learning rate** - a parameter used to optimize the model and its weights.

All models in our experiments are trained on the ImageNet database using the VGG-16 model. The architecture of the VGG-16 is shown on Figure 4.

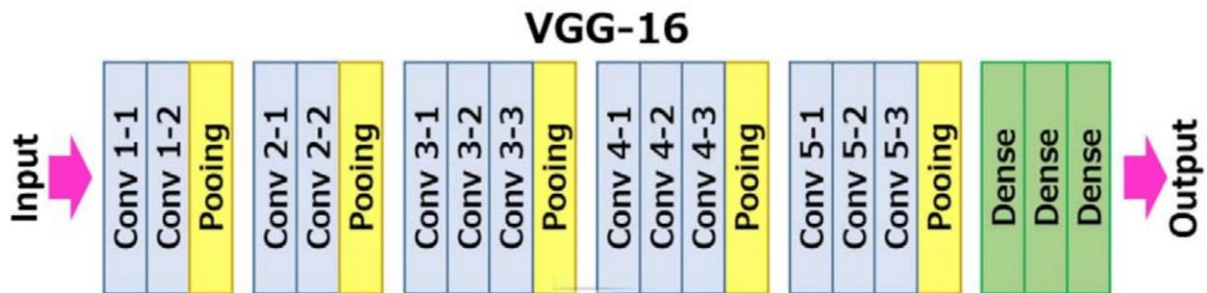


Figure 4: The layers in the VGG-16 architecture

An example of an image passing through the VGG-16 architecture is shown on Figure 4. The models were trained during 2 iterations, each consisting of 100 epochs with a batch size of 16 and a learning rate of 0.001. The dataset is split on a training, testing, and a validating set consisting of 70%, 20%, and 10% images respectively. The models do not see the testing and validating sets during training.

Figure 5 shows the basic layers of a VGG-16 architecture: the convolution, pooling, and dense layers. The convolutional layers are used to extract features by employ filters that scan an image. By doing this, the layers learn where the patterns and edges are. The pooling layers reduce the dimensions of the filters that the convolutional layers produced. Because most of the matrix that results from the pooling will be sparse, max pooling is used to keep only the information that is important for the classification. The dense layer is then used to further reduce this information to the outputs that we want to classify.

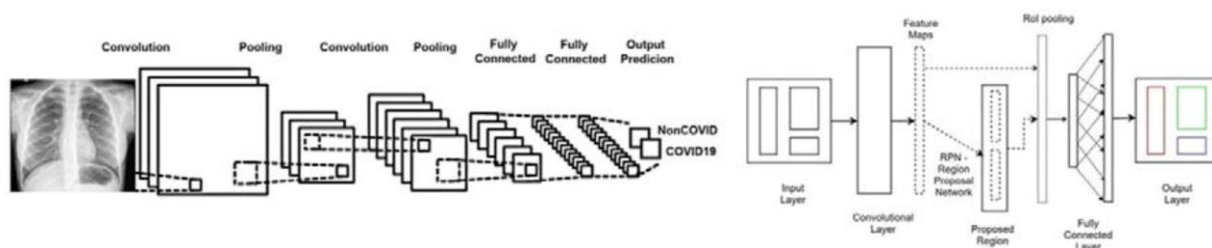


Figure 5: an image passed through the VGG-16 architecture

When we input an X-ray image to this network, what the input layer sees is a matrix with values for each pixel. Since our images are grayscale, the pixel values will be between 0 (white) and 255 (black). The input layer first gets the pixel values and the dimensions for this matrix (for example, 256x256). Then a convolution layer goes over this initial pixel matrix with a smaller, 3x3 matrix, and finds edges, textures, and early patterns. The first pooling layer reduces the dimensionality of the resulting matrix. This process is three more times and each time more distinctive patterns are picked up, such as the patterns for a specific disease. At last, a softmax layer is used to turn the network's output to a probability score that the classifier uses to determine whether there is a diseases and which one it may be.

VI. Results and Discussion

The results for all 4 experiments are described below. Figure 6 shows sample X-ray images for a few labels: healthy (No Finding), bacterial pneumonia, viral pneumonia, and COVID-19.

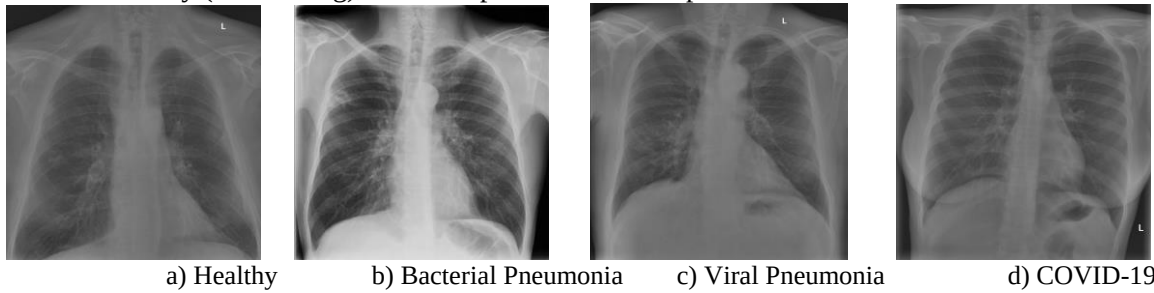


Figure 6: image examples for some of the diseases in the dataset: a) Healthy, b) Bacterial Pneumonia, c) Viral Pneumonia, d) COVID-19

VI.1 - Experiment 1

For the first experiment, because we have many more pathological images than healthy, we did two versions of the experiment. We recoded the labels and we created 2 outcomes: a normal image (No Finding), or an image with pathology (Pneumonia). Figure 7 shows that the model from the first version of the experiment achieved an **accuracy** of 49%, an **F1-score** of 0.55, a sensitivity of 0.99, and a specificity of 0.27, shown on Table 8. The low accuracy is due to the imbalanced dataset - we have about twice as many pneumonia images as healthy.

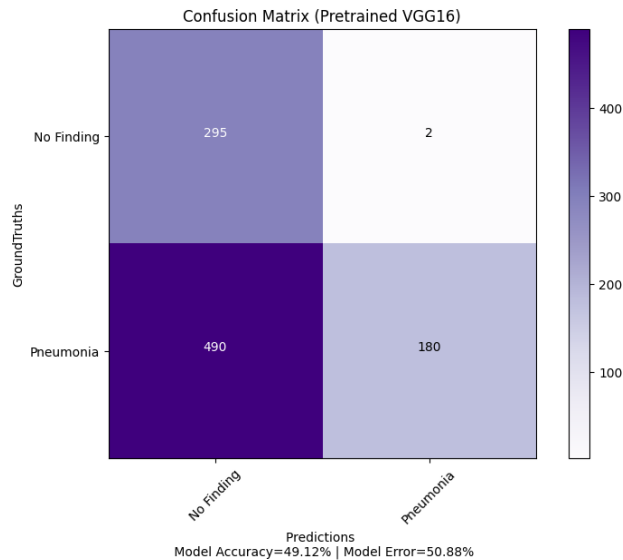


Figure 7: Binary model: No Finding, Pneumonia Confusion Matrix, Imbalanced Dataset

Class	Precision	Recall	F1-Score
No Finding	0.38	0.99	0.55
Pneumonia	0.99	0.27	0.42

Table 8: Per class metrics for the binary model

Sensitivity	Specificity
0.99	0.27

When we remove this imbalance in the second version of this experiment, the **accuracy** improves to 94.28%, as shown on Figure 8.

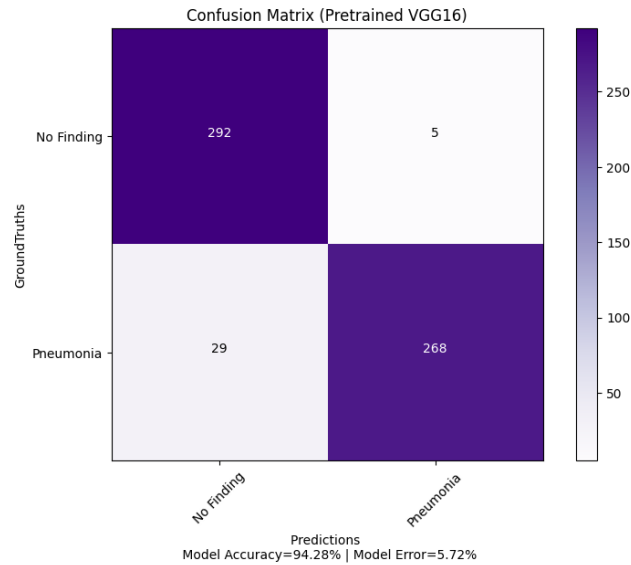


Figure 8: Binary model: No Finding, Pneumonia Confusion Matrix, Balanced Dataset

VI.2 - Experiment 2

The second experiment is a multiclass classification model differentiating between COVID-19 positive images, viral and bacterial pneumonia, and healthy images. This model achieved an **accuracy** of 84.17%, an **F1-score** of 84, as shown on Figure 9, a sensitivity of 0.0., and a specificity of 1.0. Because we only have about 72 X-Ray images for bacterial pneumonia, the model does not focus on this outcome. This can also be noted in the ROC graph in Figure 10. The first experiment shows an ability to classify COVID-19 X-Ray images, however, it cannot easily classify viral pneumonia outcomes. We can see that the model hesitates between viral pneumonia and COVID-19.

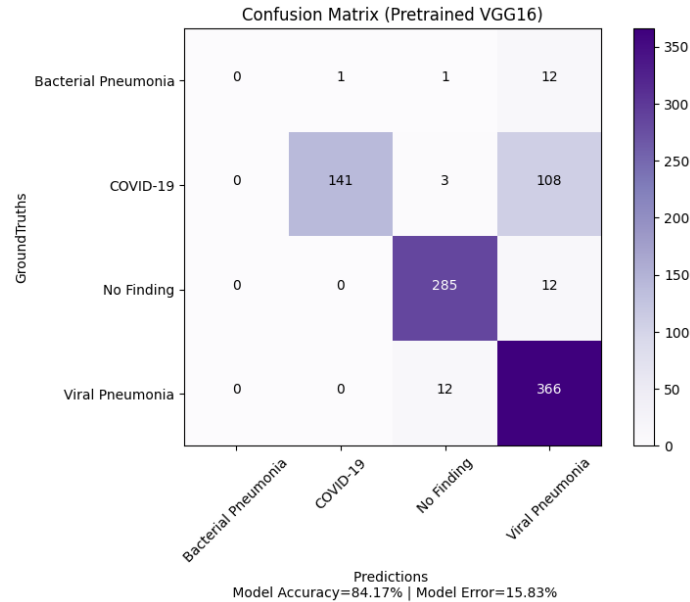


Figure 9: Multi-classifier model: Bacterial Pneumonia, COVID-19, No Finding, Viral Pneumonia

Issues with the Bacterial Pneumonia label can be seen on the ROC curve for this model, shown on Figure 10. The ROC curve shows what would happen to the true positive rate (the actual label and the predicted label match) and the false positive rate. For example, we could choose the threshold to be 0.5, 0.7, etc. If we remove the bacterial pneumonia outcome and fix the data imbalance, the model performs better - it achieves an **accuracy** of 85.76%, a sensitivity of 0.0, and a specificity of 1.0, as shown on Table 9. This version of the first experiment shows the ability to differentiate COVID-19 from viral pneumonia, as shown on Figure 11.

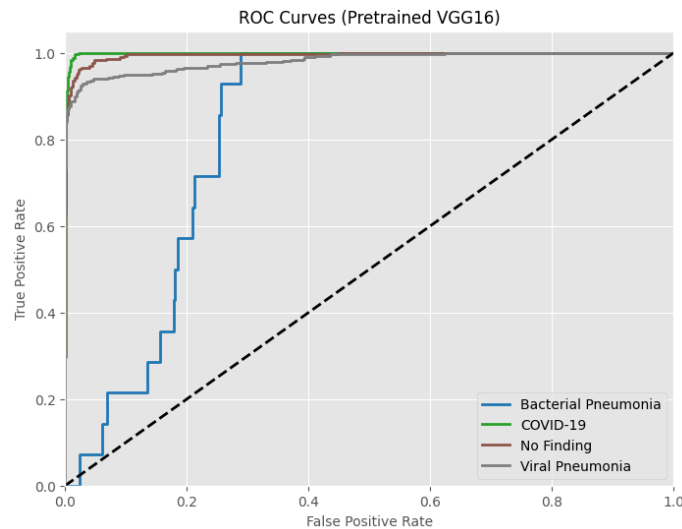


Figure 10: ROC Curve shows issues for Bacterial Pneumonia

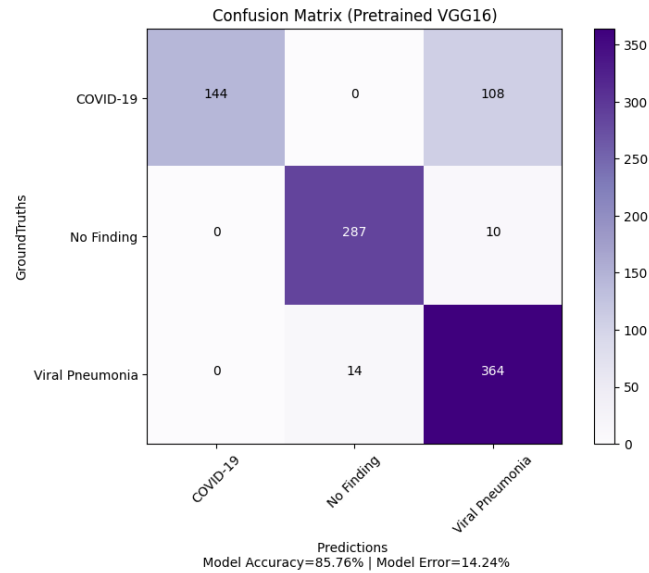


Figure 11: Multi-classifier model: COVID-19, No Finding, Viral Pneumonia

Class	Precision	Recall	F1-Score
Bacterial Pneumonia	0.0	0	0
COVID-19	1.0	0.56	0.72
No Finding	1.0	0.96	0.95
Viral Pneumonia	0.7	0.97	0.84

Table 9: Per class metrics for multi-classifier model

Sensitivity	Specificity
0.0	1

VI.3 - Experiment 3

For the third experiment, we recoded the labels again and created 4 outcomes: bacterial, fungal, lipoid, and viral pneumonia. Figure 12 shows the confusion matrix and the accuracies for this experiment. Even though the model showed a very high **accuracy** of 96.63% and an **F1-score** of 0.97, Table 10 shows a sensitivity of 1.0 and the specificity is unknown. This and the confusion matrix show that the model overfits for viral pneumonia severely. This is due to the dataset being imbalanced and the underrepresentation of bacterial, fungal, and lipoid pneumonia.

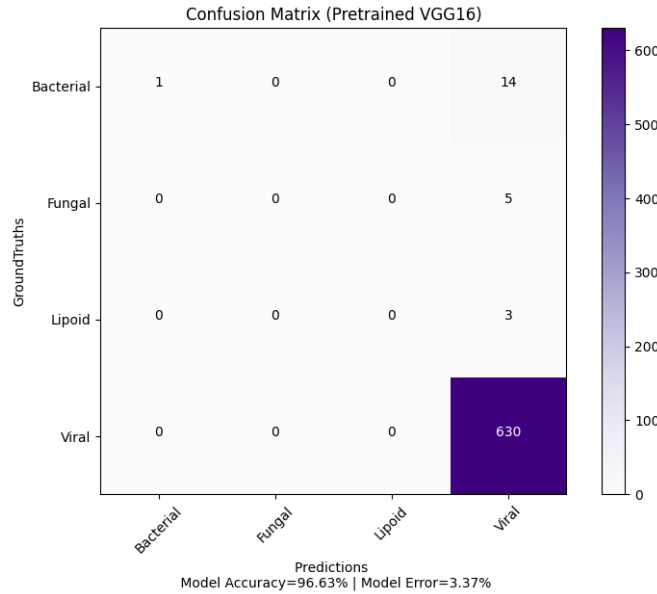


Figure 12: Multi-classifier model: Bacterial, Fungal, Lipoid, and Viral Pneumonia

Table 10: Per class metrics for multi-classifier model

Class	Precision	Recall	F1-Score
Bacterial Pneumonia	1.0	0.07	0.12
Fungal Pneumonia	0.0	0	0
Lipoid Pneumonia	0.0	0	0
Viral Pneumonia	1.0	1	0.98

Sensitivity	Specificity
1.0	nan

VI.4 - Experiment 4

The fourth experiment is similar to the first experiment in terms of data imbalances, so we created two versions for the fourth experiment as well.

For the first version, we recoded the labels and created 6 outcomes: COVID-19, influenza, MERS, SARS, varicella, and viral pneumonia. The confusion matrix is shown on Figure 13 and the accuracies are shown on Table 11. Even though the model showed a very high **accuracy** of 97.79%, an **F1-score** of 0.98, a sensitivity of 1.0, and a specificity of 0.0 shown on Table 11, because the dataset is highly imbalanced, the model only sees 2 outcomes: COVID-19 and viral pneumonia. Nevertheless, the model is able to differentiate between COVID-19 and viral pneumonia.

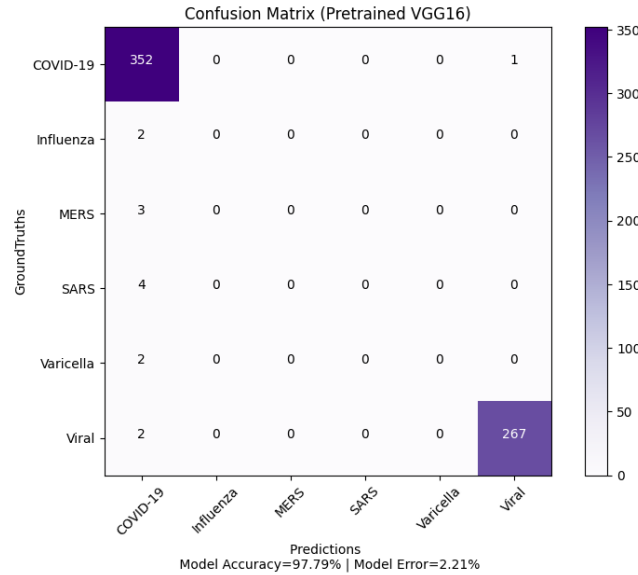


Figure 13: Multi-classifier model: COVID-19, Influenza, MERS, SARS, Varicella, Viral Pneumonia

Table 11: Per class metrics for multi-classifier model

Class	Precision	Recall	F1-Score
COVID-19	1.0	1	0.98
Influenza	0.0	0	0
MERS	0.0	0	0
SARS	0.0	0	0
Varicella	0.0	0	0
Viral Pneumonia	1.0	0.98	0.97

Sensitivity	Specificity
1.0	0

To remove this data imbalance, we created a second version of this experiment, recoded the labels, and created 2 outcomes: COVID-19 and other pneumonia. The confusion matrix and the accuracies are shown on Figure 14. This version of the experiment performs just as same, and demonstrates that if an outcome is underrepresented in the dataset, the deep learning model will simply ignore it.

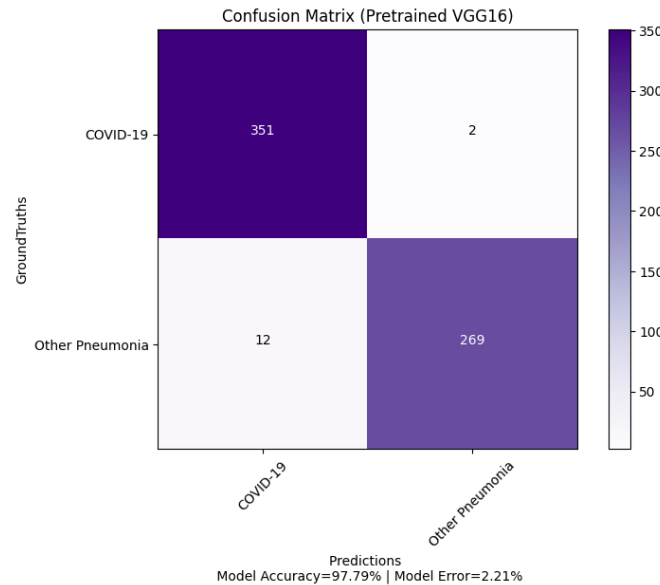


Figure 14: Multi-classifier model: COVID-19, Other Pneumonia

VII - Conclusion

The results of the four experiments we carried out demonstrate three serious issues: the first one is the restricted access to X-ray images that have been labeled correctly; the second one is the highly imbalanced distribution of the patients whose images we have access to; and the third one are the challenges that AI runs into when it is applied to low quality data or imbalanced. While deep learning can be used to discover pathology on X-ray images, this seems to have limited usefulness for the medical field, especially during a pandemic. However, once the models are asked to differentiate between diseases that may appear similarly in an X-ray, the models struggle to achieve accurate predictions. Without a sufficiently large dataset, the artificial intelligence solutions we build may not be usable in a clinical setting because of lack of covariates, like race, ethnicity, age, socio-economic status, presence or absence of symptoms, etc.

The current experiment framework is a solid base for our future research by building a set of hierarchical methods that classify different levels of pathology from X-ray images. In our next work, we are going to adjust the topology of the labels and investigate these hierarchical methods further, so we can further understand what kind of obstacles computer vision runs into when it miss-classifies lung pathology and remove them so we can alleviate potential health crises in the future.

References

1. Hasan MK, Alam MA, Dahal L, Elahi MTE, Roy S, Wahid SR, et al., 2020. Challenges of Deep Learning Methods for COVID-19 Detection Using Public Datasets, *medRxiv*
2. Tartaglione E, Barbano CA, Berzovini C, Calandri M, Grangetto M., 2020. Unveiling COVID-19 from CHEST X-Ray with Deep Learning: A Hurdles Race with Small Data. *Int J Environ Res Public Health*.
3. Burlacu A, Crisan-Dabija R, Popa IV, Artene B, Birzu V, Pricop M, et al., 2020. Curbing the AI-induced enthusiasm in diagnosing COVID-19 on chest X-Rays: the present and the near-future. *medRxiv*
4. Sohail S., 2020. Radiology of COVID-19 - Imaging the pulmonary damage. *J Pak Med Assoc*.
5. Wong HYF, Lam HYS, Fong AH, Leung ST, Chin TW, Lo CSY, et al., 2020. Frequency and Distribution of Chest Radiographic Findings in COVID-19 Positive Patients. *doi: 10.1148/radiol.2020201160*

6. Rubin GD, Ryerson CJ, Haramati LB, Sverzellati N, Kanne JP, Raoof S, et al., 2020. The Role of Chest Imaging in Patient Management during the COVID-19 Pandemic: A Multinational Consensus Statement from the Fleischner Society., *doi: 10.1016/j.chest.2020.04.003*.
7. Lai CC, Shih TP, Ko WC, Tang HJ, Hsueh PR., 2020. Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and coronavirus disease-2019 (COVID-19): The epidemic and the challenges., *doi: 10.1016/j.ijantimicag*
8. Sethy PK, Behera SK. Detection of Coronavirus Disease (COVID-19) Based on Deep Features. 2020, *doi: 10.20944/preprints202003.0300.v1*
9. COVID-19 Image Data Collection: Prospective Predictions Are the Future, Joseph Paul Cohen and Paul Morrison and Lan Dao and Karsten Roth and Tim Q Duong and Marzyeh Ghassemi, arXiv:2006.11988, <https://github.com/ieee8023/covid-chestxray-dataset>, 2020
10. COVID-19 Chest X-Ray Dataset Initiative, Wang L., Wong A., Qui Lin Zh., McInnis P., Chung A., Gunraj H., Lee J., Ross M., VanBerlo B., Ebadi A., Git KA., AL-Haimi A., <https://github.com/agchung/Figure1-COVID-chestxray-dataset>
11. Actualmed COVID-19 Chest X-Ray Dataset Initiative, Wang L., Wong A., Qui Lin Zh., McInnis P., Chung A., Gunraj H., Lee J., Ross M., VanBerlo B., Ebadi A., Git KA., AL-Haimi A., <https://github.com/agchung/Actualmed-COVID-chestxray-dataset>
12. M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz, M. T. Islam, "Can AI help in screening Viral and COVID-19 pneumonia?" IEEE Access, Vol. 8, 2020, pp. 132665 - 132676, <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>
13. Shibly KH, Dey SK, Islam MT, Rahman MM. COVID faster R-CNN: A novel framework to Diagnose Novel Coronavirus Disease (COVID-19) in X-Ray images., *doi: 10.1016/j.imu.2020.100405*
14. Busby LP, Courtier JL, Glastonbury CM, 2018. Bias in radiology: the how and why of misses and misinterpretations. Radiographics., *doi: 10.1148/rq.2018170107*
15. Farhat H, Sakr GE, Kilany R., 2020. Deep learning applications in pulmonary medical imaging: recent updates and insights on COVID-19. *doi: 10.1007/s00138-020-01101-5*
16. Mangal A, Kalia S, Rajgopal H, Rangarajan K, Namboodiri V, Banerjee S, et al., 2020. CovidAID: COVID-19 Detection Using Chest X-Ray., *doi: 10.48550/arXiv.2004.09803*
17. Stogiannos N, Fotopoulos D, Woznitza N, Malamateniou C., 2020. COVID-19 in the radiology department: What radiographers need to know., *doi: 10.1016/j.radi.2020.05.012*
18. Heinrichs B, Eickhoff SB., 2020. Your evidence? Machine learning algorithms for medical diagnosis and prediction., *doi: 10.1002/hbm.24886*
19. Mobiny A, Cicalese P, Zare S, Yuan P, Abavisan M, Wu C, et al., 2020. Radiologist-Level COVID-19 Detection Using CT Scans with Detail-Oriented Capsule Networks., *doi: 10.48550/arXiv.2004.07407*
20. Tahir AM, Qiblawey Y, Khandakar A, Rahman T, Khurshid U, Musharavati F, Islam MT, Kiranyaz S, Al-Maadeed S, Chowdhury MEH, 2021. Learning for Reliable Classification of COVID-19, MERS, and SARS from Chest X-Ray Images, *doi: 10.1007/s12559-021-09955-1*
21. Pooch, E., Ballester P., Barrow R., 2021. Can we trust deep learning models diagnosis? The impact of domain shift in chest radiograph classification, *doi: 10.48550/arXiv.1909.01940*

Comparison of threading and non-threading implementations in the Pascal Triangle

Agim Rushiti¹, Rilind Kaba², Shpetim Rexhepi²

Email(s): agim_rushiti@yahoo.com, rilind.kaba@students.unt.edu.mk, shpetim.rexhepi@unt.edu.mk

¹Ministry of Science and Education, North Macedonia

²Faculty of Informatics “Mother Teresa” University Skopje, North Macedonia

Abstract

Coefficients of binomial expression are calculated in the Pascal Triangle. Finding the coefficients in Pascal's triangle is done in such a way that to find the coefficients of a certain row we need the previous row, which means to find a row n we need the coefficients of row $n - 1$ in Pascal's triangle. The linear calculation of Pascal's triangle takes a lot of time, for this reason, this paper will analyze and compare the linear solution of the problem and the parallel solution.

This paper presents a comparative analysis of linear and parallel implementations in computing Pascal's Triangle coefficients. Pascal's Triangle, a foundational mathematical construct, is pivotal in calculating binomial coefficients. Traditional linear computation methods are contrasted with parallel approaches to assess their respective efficiency in time, processor utilization, and memory consumption. Our study unveils intriguing insights into the performance dynamics of both methodologies through meticulous experimentation conducted on a standard computing platform. The linear approach, while straightforward, exhibits exponential time complexity as the dataset size escalates. Conversely, parallel programming, leveraging threading techniques, showcases promising efficiency gains, particularly with larger datasets. We delve into the intricacies of thread allocation and workload distribution, shedding light on the optimal utilization of computational resources. Furthermore, our investigation elucidates the scalability of each approach, shedding light on their performance under varying dataset sizes. We discuss the implications of our findings for computational efficiency and resource utilization in real-world applications.

Keywords: Pascal Triangle, threading, non-threading, linear solution, parallel solution

Introduction

In the introduction part we use the terminology, notations and remarks used in [4], [6], [7] and [8]. Consider the numerical sequence $d_0, d_1, d_2, \dots, d_n$ such that $n = 1, 2, 3, \dots, n$ (for $n = 0$ we get d_0). The new numerical sequence $s_0, s_1, s_2, \dots, s_{n+1}$, can be generated using the following rules.

$$s_0 = d_0 \quad (1.1)$$

$$s_k = d_{k-1} + d_k \quad (1 \leq k \leq n) \quad (1.2)$$

$$s_{n+1} = d_n \quad (1.3)$$

The French mathematician Blaise Pascal (1623-1662) began to investigate the properties of a triangular table of numbers, such that each row follows from the previous relations (1.1) - (1.3). This table, which we will examine further, is known as "Pascal's Triangle".

Remark 1: Using relations (1.1)- (1.3) we get:

$$\begin{aligned} s_0 + s_1 + s_2 + s_3 + \dots + s_n + s_{n+1} \\ = d_0 + (d_0 + d_1) + (d_1 + d_2) + \dots + (d_{n-1} + d_n) + d_n \\ = (d_0 + d_0) + (d_1 + d_1) + (d_2 + d_2) + \dots + (d_n + d_n) \\ = 2(d_0 + d_1 + \dots + d_n) \end{aligned} \quad (1.4)$$

Remark 2: The numerical series $d_0 + d_1 + \dots + d_n$ is symmetric if for every integer k from 0 to n , holds the relation

$$d_k = d_{n-k} \quad (1.5)$$

For example, the sequence of four numbers 1,0,0,1 is symmetric. The series $s_0 + s_1 + \dots + s_{n+1}$ of numbers obtained from the symmetric series $d_0 + d_1 + \dots + d_n$ from the relations of Pascal's triangle is symmetric to itself. To prove this, we must check validity of the following equality,

$$s_k = s_{(n+1)-k} \quad (1.6)$$

For $k = 0, 1, 2, \dots, n+1$. But for $k = 0$ and $k = n+1$ equation (1.6) follows from relations (1.1) and (1.3) and the condition $d_0 = d_n$ (which follows from (1.5) for $k = 0$). For $(1 \leq k \leq n)$, we have:

$$\begin{aligned} s_k = d_{k-1} + d_k = d_{n-(k-1)} + d_{n-k} = d_{(n+1)-k} + d_{[(n+1)-k]-1} \\ = d_{[(n+1)-k]-1} + d_{(n+1)-k} = s_{(n+1)-k} \end{aligned} \quad (1.7)$$

In the case of our example, applying Pascal's relations to the array 1 0 0 1, gives the array of five elements 1 1 0 1 1, that is symmetric to itself. Let's consider the case when the sequence contains the only number 1. We can call this "Pascal zero sequence ". From here, with the help of Pascal's relations, the sequence can be generated, which we can call Pascal's first sequence. By applying Pascal's sequence, can be generated the second Pascal's sequence from the first Pascal's sequence, and so on.

From remarks 1 and 2 can be concluded:

1. The sum of the numbers n -th in Pascal's sequence is 2^n (the sum of the zero sequence is $2^0 = 1$).
2. All sequences of Pascal are symmetric (also the zero sequence is taken symmetric).

Let us write Pascal's series, one after the other, so that each number of each series is between and below those numbers of the previous line from which it is calculated. We obtain an infinite table called Pascal's triangle, which fills the interior of an angle of each of its segments, consisting of the first row to the n -th row that forms the triangle. The classic configuration of Pascal's triangle can be described as a triangle table of numbers, where each number is equal to the sum of the two numbers above it [5,9]. The segment of Pascal's triangle containing the first 15 lines (from zero to the fifteenth) is shown in the following Figure.

[illegible]

Pascal triangle (Source: Authors' own elaboration)

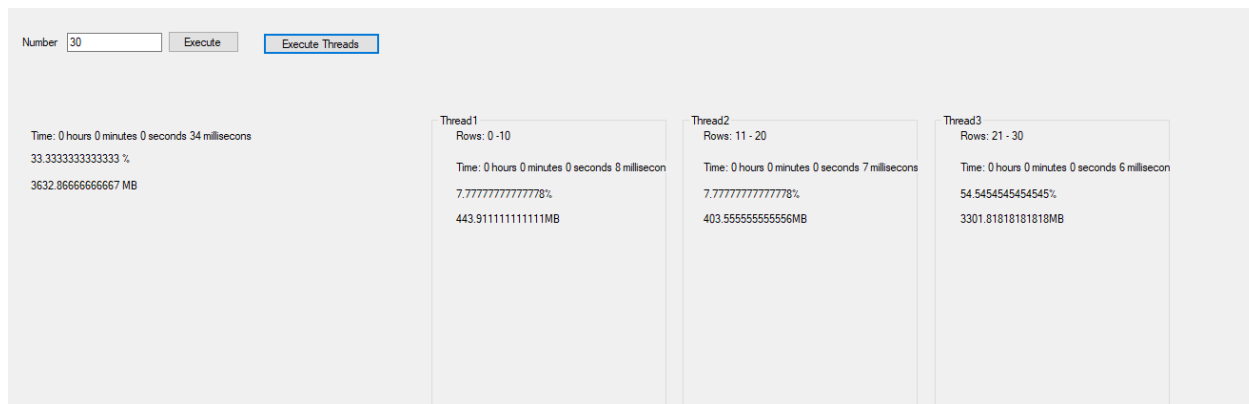
Pascal's Triangle gives us the coefficients of binomial expression.

In [1] is introduced $\binom{n}{r} = \frac{n!}{(n-r)!r!}$

Example:

$$(1-x)^3 = \binom{3}{0}x^0 - \binom{3}{1}x^1 + \binom{3}{2}x^2 - \binom{3}{3}x^3 = 1 - 3x + 3x^2 - x^3$$

The implementation of the program for linear and parallel solution of Pascal's triangle is a desktop application in the C# language. Program gives us data's for how much time is spent, how much processor is used and how much memory is consumed.



The experiments will be performed on a computer with these performances:

- Processor - Intel(R) Core(TM) i5-4300 CPU @ 1.90GHz
- Memory - 8GB

Linear implementation – Non-Threading

In [2] is introduced:

```
int no_row,c=1,blk,i,j;

Console.Write("\n\n");
Console.Write("Display the Pascal's triangle:\n");
Console.Write("-----");
Console.Write("\n\n");

Console.Write("Input number of rows: ");
no_row = Convert.ToInt32(Console.ReadLine());
for(i=0;i<no_row;i++)
{
    for(blk=1;blk<=no_row-i;blk++)
        Console.Write(" ");
    for(j=0;j<=i;j++)
    {
        if (j==0||i==0)
            c=1;
        else
            c=c*(i-j+1)/j;
        Console.Write("{0} ",c);
    }
    Console.Write("\n");
}
```

Number of rows	Time spent	Processor consumed (%)	Memory spent (MB)
10	8 ms	30	3423.07
100	1 sec 783 ms	63.11	3391.4
300	55 sec 196 ms	48.20322	3469.74
500	8 min 16 sec 673 ms	44.51	3506.9

As we can see here from the above table the time spent increases exponentially with the increasing number of rows. Memory spent increases gradually while Processor consumed doesn't change much.

Parallel Implementation – Threading

The program is designed to work with 3 Threads, of which 2 Threads execute 33% of each row and 1 Thread 34%. Threads start executing at the same time.

Example: If we want 100 lines from Pascal's triangle, the division is done in this way:

1. Thread - 1 -> 0 – 33 rows
2. Thread - 2 -> 34 – 66 rows
3. Thread - 3 -> 67 – 100 rows

Example :

$$\text{Third row} = \frac{3!}{(3-0)!0!} \quad \frac{3!}{(3-1)!1!} \quad \frac{3!}{(3-2)!2!} \quad \frac{3!}{(3-3)!3!}$$

All the coefficients of the row are found through the formula

$$\binom{n}{r} = \frac{n!}{(n-r)!r!}.$$

Thread 1

```
double cpu = 0, memory = 0;
Stopwatch stopWatch = new Stopwatch();

string row = "";
string pascalTriangle = "";

PerformanceCounter cpuCounter;
PerformanceCounter ramCounter;

cpuCounter = new PerformanceCounter("Processor", "% Processor Time", "_Total");
ramCounter = new PerformanceCounter("Memory", "Available MBytes");

stopWatch.Start();
for (int j = 0; j <= (int)no_row / 3; j++)
{
    row = "";
    for (double i = 0; i < j; i++)
    {
        row = row + "  " + (factorial(j - 1) / ((factorial(i) * factorial((j - 1) - i))));
    }

    pascalTriangle = pascalTriangle + " \n " + row;
    cpu = cpu + cpuCounter.NextValue();
    memory = memory + ramCounter.NextValue();
}
stopWatch.Stop();

Thread1Rows.Invoke((MethodInvoker)((() => Thread1Rows.Text = "Rows: 0 -" + (int)no_row / 3)));

Thread1Cpu.Invoke((MethodInvoker)((() => Thread1Cpu.Text = cpu / (int)no_row / 3 + "%")));
Thread1Memory.Invoke((MethodInvoker)((() => Thread1Memory.Text = memory / (int)no_row / 3 + "MB")));

TimeSpan timeSpan = stopWatch.Elapsed;

Thread1Time.Invoke((MethodInvoker)((() => Thread1Time.Text = "Time: " + timeSpan.Hours + " hours " + timeSpan.Minutes + " minutes " + timeSpan.Seconds + " seconds " + timeSpan.Milliseconds + " milliseconds")));
```

Thread 2

```
double cpu = 0, memory = 0;
Stopwatch stopWatch = new Stopwatch();

string row = "";
string pascalTriangle = "";

PerformanceCounter cpuCounter;
PerformanceCounter ramCounter;

cpuCounter = new PerformanceCounter("Processor", "% Processor Time", "_Total");
ramCounter = new PerformanceCounter("Memory", "Available MBytes");

stopWatch.Start();

for (int j = ((int)no_row / 3) + 1; j <= ((int)no_row / 3) * 2; j++)
{
    row = "";
    for (double i = 0; i < j; i++)
    {
        row = row + " " + (factorial(j - 1) / ((factorial(i) *
            factorial((j - 1) - i))));
    }

    pascalTriangle = pascalTriangle + " \n " + row;
    cpu = cpu + cpuCounter.NextValue();
    memory = memory + ramCounter.NextValue();
}

stopWatch.Stop();

Thread2Rows.Invoke((MethodInvoker)((() => Thread2Rows.Text = "Rows: " + (((int)no_row
/ 3) + 1) + " - " + ((int)no_row / 3) * 2));

Thread2Cpu.Invoke((MethodInvoker)((() => Thread2Cpu.Text = cpu / (int)no_row / 3 +
"%"));

Thread2Memory.Invoke((MethodInvoker)((() => Thread2Memory.Text = memory /
(int)no_row / 3 + "MB"));

TimeSpan timeSpan = stopWatch.Elapsed;
Thread2Time.Invoke((MethodInvoker)((() => Thread2Time.Text = "Time: " + timeSpan.Hours
+ " hours " + timeSpan.Minutes + " minutes " + timeSpan.Seconds + " seconds " +
timeSpan.Milliseconds + " millisecons"));
```

Thread 3

```
double cpu = 0, memory = 0;
Stopwatch stopWatch = new Stopwatch();

string row = "";
string pascalTriangle = "";
```

```

PerformanceCounter cpuCounter;
PerformanceCounter ramCounter;

cpuCounter = new PerformanceCounter("Processor", "% Processor Time", "_Total");
ramCounter = new PerformanceCounter("Memory", "Available MBytes");

stopWatch.Start();

for (int j = (((int)no_row / 3) * 2) + 1; j <= no_row; j++)
{
    row = "";
    for (double i = 0; i < j; i++)
    {
        row = row + "  " + (factorial(j - 1) / ((factorial(i) * factorial((j - 1) -
i)))));
    }

    pascalTriangle = pascalTriangle + " \n " + row;
    cpu = cpu + cpuCounter.NextValue();
    memory = memory + ramCounter.NextValue();
}

stopWatch.Stop();

Thread3Rows.Invoke((MethodInvoker)((() => Thread3Rows.Text = "Rows: " + (((int)no_row
/ 3) * 2) + 1) + " - " + no_row));

Thread3Cpu.Invoke((MethodInvoker)((() => Thread3Cpu.Text = cpu / (((int)no_row / 3) +
1) + "%"));

Thread3Memory.Invoke((MethodInvoker)((() => Thread3Memory.Text = memory /
(((int)no_row / 3) + 1) + "MB"));

TimeSpan timeSpan = stopWatch.Elapsed;

Thread3Time.Invoke((MethodInvoker)((() => Thread3Time.Text = "Time: " +
timeSpan.Hours + " hours " + timeSpan.Minutes + " minutes " + timeSpan.Seconds + "
seconds " + timeSpan.Milliseconds + " millisecons"));

```

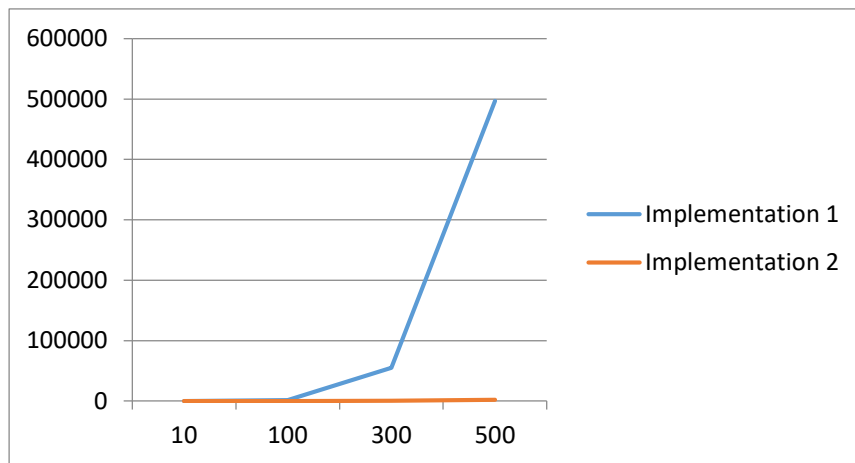
Number of rows	Time spent (average 3 threads)	Processor consumed (%) (average 3 threads)	Memory spent (MB) (average 3 threads)
10	8 ms	18.3	1490
100	36 ms	29.46	1501.43
300	393 ms	24.38	1496.431
500	2 sec 201 ms	18.16	1515.333

As we can see here the time spent increases with the increasing number of rows but increase is gradually. Memory spent is constant.

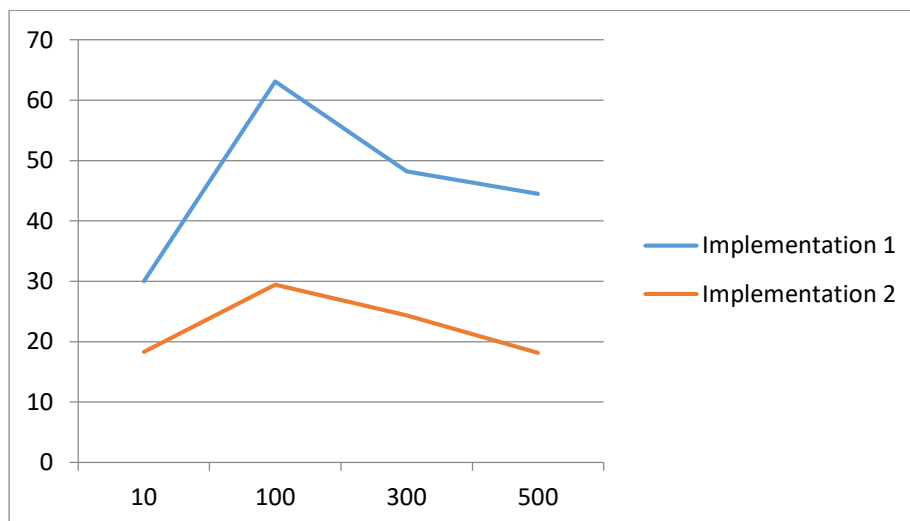
Linear Solution – Non-threading vs Parallel Solution - Threading

From the experiments done in the program at the time level, we see that the linear non-threading implementation has an exponential increase in time compared to the parallel Threading implementation, making it less efficient due to the long execution time. From the results, we can see that for the number of lines 300, the big time difference begins, where the linear implementation has executed for about 55 seconds, while the parallel implementation for 393 milliseconds.

Time spent

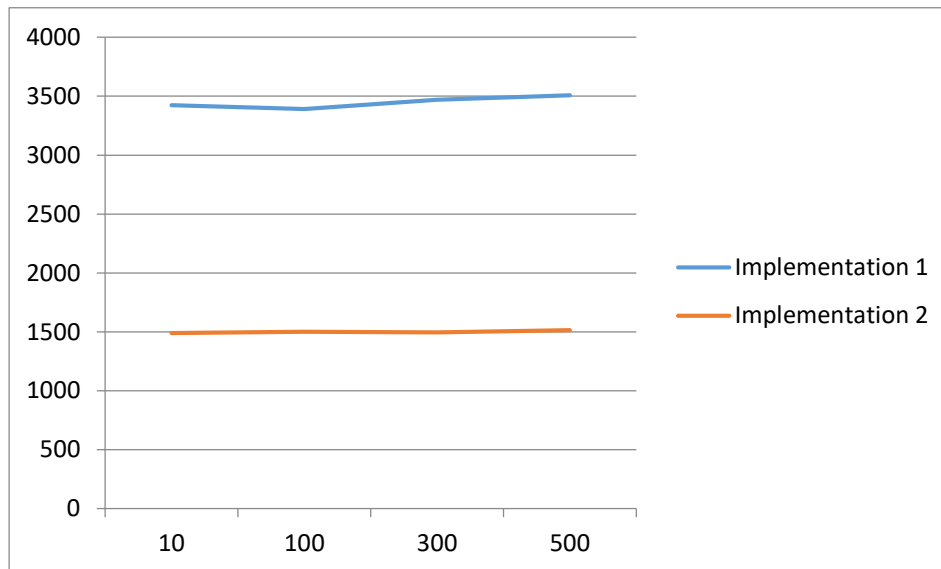


Processor spent (%)



From the above graph we can see that we have an increase in the percentage of processor consumption for 100 rows in both implementations, then there is a gradual decrease.

Memory spend (MB)



From the graph above we can see that both implementations have a constant memory consumption.

From the experiments done in the program for 10 rows, we have the same execution time, also the processor consumption is lower in implementation 2 - Threading, we have a difference in memory consumption, where implementation 1 – Non-threading consumes almost 2 times more memory than implementation 2 - Threading. For a smaller number of rows, we see that implementation 2- Threading is more efficient as it consumes less resources to obtain the result.

10 rows	Non-Threading	Threading
Time spent	8ms	8ms
Processor consumed %	30	18.3
Memory spent MB	3423	1490

Conclusion

The difference between linear and parallel programming can be seen in the larger amount of data that must be processed. Parallel programming is more efficient since it divides a given problem into smaller parts and executes them in parallel while linear programming is executed one by one that makes it less efficient. Parallel programming for large numbers of data is much more efficient as the execution time is many times faster than linear programming. Parallel computing as a message passing library accommodates a natural and easy partitioning of the problem, it provides portability and efficiency, and it has received wide acceptance by academia and industry [10].

This comparative analysis between linear and parallel implementations in computing Pascal's Triangle coefficients illuminates the transformative potential of parallel programming paradigms. Through

meticulous experimentation and analysis, we have demonstrated that parallel approaches offer significant advantages in terms of computational efficiency, particularly when confronted with larger datasets.

The exponential time complexity observed in linear methods underscores the urgent need for scalable solutions, especially in mathematical computation. Parallel programming, leveraging threading techniques, emerges as a powerful tool for tackling complex computational tasks, offering substantial reductions in processing time and resource utilization.

Our findings reaffirm the importance of parallel computing in enhancing computational performance and highlight its broader implications for diverse fields reliant on mathematical analysis. From scientific simulations to data processing tasks, the efficiency gains afforded by parallel programming can revolutionize how we approach complex computational problems.

Looking ahead, future research endeavors could delve deeper into optimizing thread allocation strategies and exploring novel approaches to exploit hardware parallelism more effectively. Additionally, investigating the integration of parallel computing techniques into existing computational frameworks holds promise for advancing the frontiers of mathematical analysis and computational science.

Moreover, exploring the impact of parallelization techniques, such as GPU acceleration or distributed computing, could further enhance computational efficiency and scalability. Furthermore, examining the applicability of parallel programming methodologies to other mathematical problems beyond Pascal's Triangle could unveil new avenues for exploration and innovation.

Our study underscores the transformative potential of parallel programming paradigms, offering valuable insights for researchers, practitioners, and educators alike. By harnessing the power of parallelism and charting new paths for future research, we can unlock new realms of computational efficiency and pave the way for groundbreaking advancements in scientific and mathematical inquiry.

References

- [1] Weisstein, Eric.W. "Pascal's triangle" <https://mathworld.wolfram.com> /(2002)
- [2] w3resource. [Online] <https://www.w3resource.com/csharp-exercises/for-loop/csharp-for-loop-exercise-33.php>.
- [3] Markmann, A. An Exercise in Parallel Programming The Pascal Triangle.(2009) https://ursula.chem.yale.edu/~batista/personal/Andreas/side_projects/omp_exercise.pdf
- [4] Aliu, A., Rexhepi, S., & Iseni, E. (2023). Efficiency of understanding some mathematical problems by means of Pascal's triangle. *International Electronic Journal of Mathematics Education*, 18(4), em0753.
- [5] Hosch, W. L. (2013). Pascal's triangle. <https://www.britannica.com/science/Pascals-triangle>
- [6] Jerphagnon, L., & Orcibal, J. (2020). Blaise Pascal. <https://www.britannica.com/biography/Blaise-Pascal>
- [7] Kuz'min, O. V. (2000b). Triangle and pyramid of Pascal: Properties and generalizations. *Soros. Obraz. Zh.*, 6(5), 101-109.
- [8] Uspenskiĭ, V. A. (1976). Pascal's triangle. Mir Publishers.
- [9] Gardner, M. (1974). Mathematical games. *Scientific American*. <https://doi.org/10.1038/scientificamerican0874-98>
- [10] G. E. Karniadakis dhe R. M. Kirby II, *Parallel Scientific Computing in C++ and MPI*, Cambridge University Press, 2003

Simulation and animation of a 3D avatar from a realistic human face model

Mitko Trpkoski [†], Prof. d-r Anis Sefidanoski [‡], Slavcho Mitevski [‡],

Arbnor Rushiti [‡]

Email(s): trpkoski.mitko@uklo.edu.mk, anis.sefidanoski@uist.edu.mk, slavco_mitevski@yahoo.com, ar26730@seeu.edu.mk.

[†] Department of IKT, Municipality of Ohrid, Ohrid, 6000, R.N. Macedonia

[‡] Faculty of Applied IT, Machine Intelligence and Robotics, University of Information Science and Technology (UIST) "St. Paul the Apostle", R.N Macedonia

[‡] Department of IT, Singular, Skopje, 1000, R.N Macedonia

[‡] Faculty of Contemporary Sciences and Technologies, SEEU University, Tetovo/Republic of North Macedonia

Abstract

In the early 1980s, during the presentation of films like Dreamflight or The Joggler, the first virtual 3D avatars and their animations appeared. With the research that followed until today, the development and evaluation of their appearance and their photorealistic displays in various situations and their simulation followed. Parallel to their evaluation and the more frequent use of virtual 3D avatars in order to present them photorealistically due to the perception of the human eye for details, more and more emphasis is placed on their animation and interaction in real time during the actual modeling and animating in real time. Also, with the aim of photorealistic presentation and obtaining a real feeling and representation in the 3D virtual world, a variety of techniques are being developed for automatic animation and presentation of photorealistic 3D avatars or their 3D face models that can be animated with the help of a variety of techniques for using audio. or video as inputs. By using such techniques, the end result is a photorealistic 3D virtual avatar that behaves like a real character and has the capacity to interact during a conversation. Such 3D photorealistic avatars, if they are well "trained", have the ability to express a variety of emotions that real models express in reality and to give realistic answers to a variety of questions. This paper analyzes different types of communication or interaction between real people and 3D photorealistic avatar models in real time. When using different techniques or concepts for some realized interaction, different possible display solutions are illustrated that the 3D virtual avatar displays when speaking or moving like a real-life person. Analyzes of the presentation of different expressions on photorealistic 3D models of people and the use of their speech during presentation in various types of communication (verbal or non-verbal) and thus the synchronization of the audio itself - their voice - have also been made. Today, with the use of a variety of systems or algorithms for display or interaction, 3D photorealistic avatars find great application and are used more and more often, saving performance and time.

Keywords

Virtual human, clone, facial deformation, real-time facial animation, 3-D feature points, virtual dialogue, virtual face.

1 Introduction

To clone is to photorealistically render a virtual 3D avatar on a 2D output unit. In the contexts presented in this paper, it is photorealistic to render a 3D virtual model of a real model of an existing person. In this particular case, the interest of the presentation is limited to one component of the human figure, which is the face as the most complex part of the presentation. In many cases of presentation, the passive face also conveys a large amount of information, while when it comes to photorealistic presentation and its animation, the range of motion it offers is very large, and parts such as teeth, lips, speaking tongue, eyelashes come to the fore. of eyes, head movement and other additional elements that need to be photo-realistically displayed to obtain a quality 3D display. In the case of misrepresentation of some elements such as muscles or facial lines, a photorealistic rendering that is easily noticeable is not obtained and thus the display is of poor quality.

For a photorealistic rendering of a 3D model of a cloned face, a framework should be used that describes and contains the geometric shapes and animation capabilities of the model being used (shown). Also, while working, the textures of the real model must be taken into account to make it photorealistic. Models that are static 3D models when rendered are unsuitable models when rendered due to their non-photorealistic 3D presentation and are accordingly unsuitable for our purpose in photorealistic 3D presentation. The way in which the modeling of the geometry of a 3D photorealistic avatar takes place in many cases depends on the application of the animation itself and its potential for further development. When modeling a 3D photorealistic avatar face from a real model, as many aspects as mentioned above should be observed and allow the rendered 3D model to look photorealistic and identifiable as its clone. The main prerequisites for 3D photorealistic rendering of a real face model can be its shape, movements and expression of emotions during speech. Such rendering requires the combined use of techniques from different fields of geometric modeling and animating photorealistic avatars. The modeling, shaping, and rendering of the 3D photorealistic avatar includes other visual features from the realistic model, such as skin color and texture. Also, in 3D photorealistic modeling, the shape can be taken from photographs or other data that describe the actual model in detail. Also, facial movement synchronization involves deforming its geometry during rendering according to physical or predefined rules of movements transmitted during speech and expressions of the 3D photorealistic face model (Figure 1).

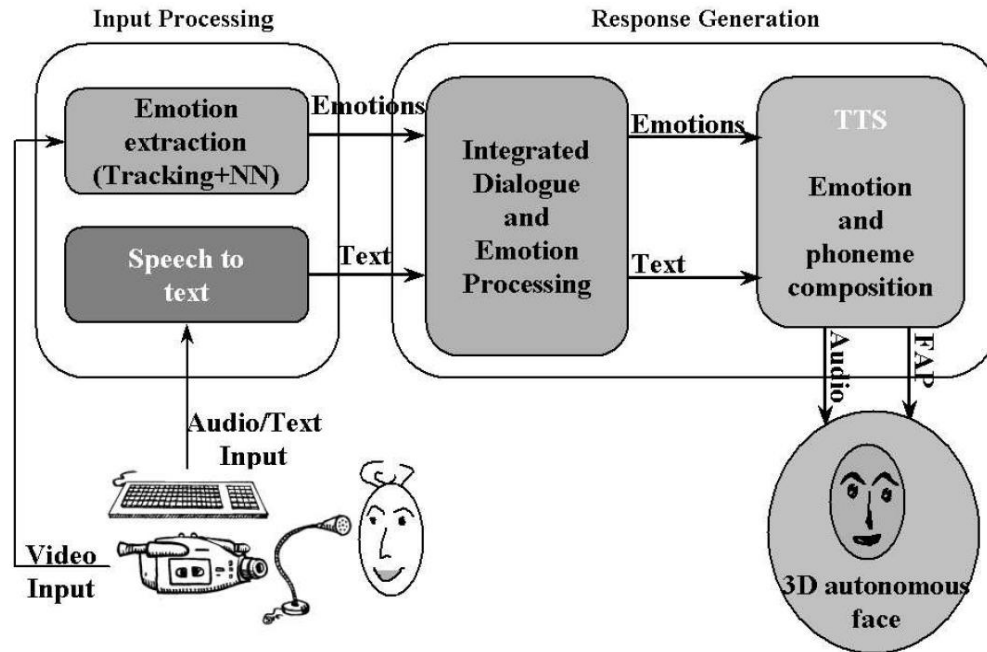


Figure 1

2 Facial communication between real and virtual 3D avatar

Early human communications and interactions with 3D virtual human avatars have long been of great interest and attention to humans. This interaction can be displayed in the system that automatically generates speech on a 3D photorealistic avatar, which also includes expressions on the face itself has multiple 3D models that are based on realistic models of human faces. But it should also be taken into account that those who are on this system do not include aspects that clone the actual model. For the purpose of photorealistic display, a special mechanism can also be used to control the action of animations, the structure of the face of the 3D model during photorealistic display in communication. Virtual and photorealistic rendering of 3D models to show what can define the parameters of the 3D model itself is used as an autonomous-independent actor in its own communication.

Today many different ways of communication in the virtual world with 3D photorealistic models are possible, of which four are the most common: communication between a real model with a virtual 3D photorealistic model from an avatar, a cloned 3D photorealistic model with another clone 3D photorealistic model, a virtual virtual 3D photorealistic model with 3D photorealistic model and cloned 3D model from real shoe with virtual 3D photorealistic model. The text that follows briefly describes each individual communication

With all the believable and photorealistic rendering it can be pre-defined and recorded and used by the autonomous 3D virtual person in order to respond through his/her facial expressions and speech from the real model. Also, when presenting as special cases of such a situation, the imitation of gestures or speech with the actual real model can be taken. This image is shown in the following image - Figure 2.

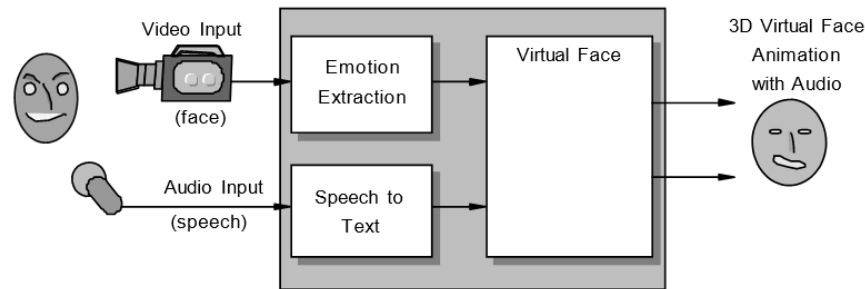


Figure 2

In precisely defined virtual 3D environments, it can be mentioned that communication representatives in different locations in the world represent themselves from their 3D environments as clone avatars. In order to present the virtual 3D avatars in this way, the 3D avatar should be modeled, which is a clone of the real model, then the face of the real model should be recorded with a camera for each student in the communication, so that they can finally be displayed and look photorealistic - real. With the aim of such a photorealistic display of all animations from the parameters taken, examples of movement from the real model are mapped to the photorealistic 3D model, and thus the speech - audio is reproduced for each person on each 3D model separately and a photorealistic 3D animation is obtained. Such communication is shown in the following figure - Figure 3.

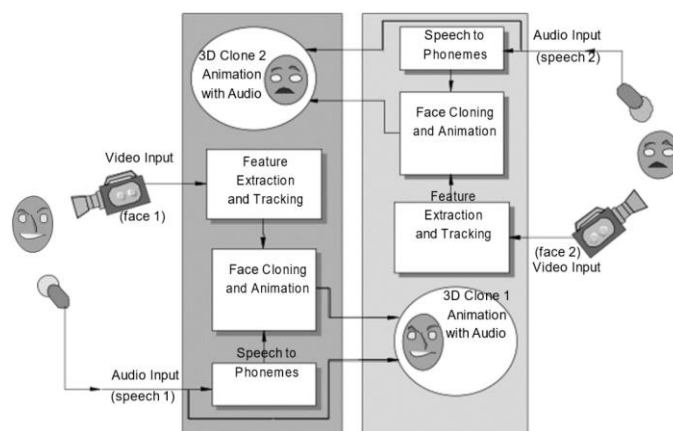


Figure 3

In the virtual 3D world, communication between an autonomous 3D virtual avatar with another 3D virtual avatar that is also autonomous and separate is also possible. Such communication may include conversations between two or more people in the conversation and the expression of some avatars who are immediate participants. Example rendering situations are not important other aspects of cloning the 3D photorealistic models in the dialog as it automatically contains enough AI and capacity to understand what the other virtual 3D realistic model is looking for and thus generate a plausible answer to the question posed.

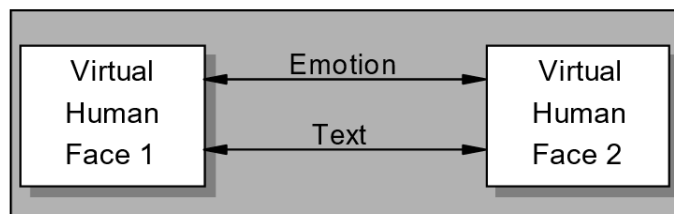


Figure 4

Virtual 3D environments that are populated by a large number of clones of real human models and virtual 3D avatars that are autonomous in that 3D virtual world require communication between a photorealistic 3D model and a model that is a clone of a real model. In order to display the above animations, several aspects are needed when determining the animation in terms of shaping and photorealistic display in order to reconstruct the 3D photorealistic model and the own models to look natural and photorealistic. In equal cases of rendering, the visual 3D model that is autonomous can have the capacity to react and use the expressions displayed on the face itself and thus contain speech in order to appear as an international and photorealistic own ation. This kind of presentation of animations and thus the realization of communication and dialogue attracts great interest among the participants and when there is more than one real student in the dialogue itself who is represented by his 3D photorealistic model - a clone or several 3D autonomous virtual avatars. Also an interesting detail is that it attracts a lot of interest among those present and the audience is when one of the persons would not be interested in seeing his 3D model – he can see the presence of other participants in the dialogue or communication itself. This communication is shown in the following image – Figure 5.

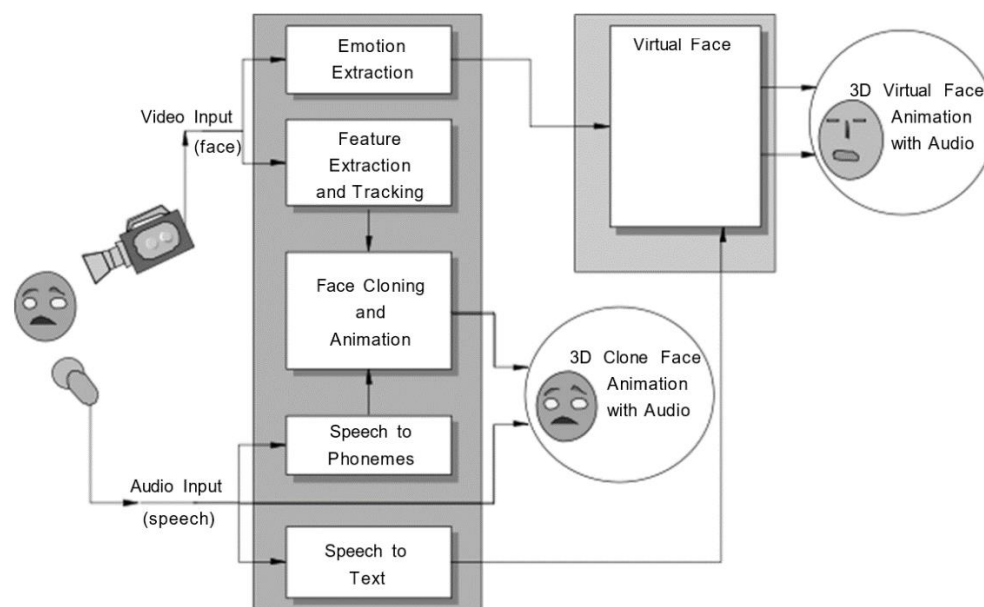


Figure 5

Realizing dialogue/communication between a clone of a real model and a virtual 3D avatar can solve all the problems associated with displaying cloned avatar models and their communication with other models, and such legacy communication/dialogue can be considered as the most general case for all the above described cases of inspection communication. The contact of this kind of display is useful for especially photorealistic display of the most common systems that are used are concentrated on one detail that is chosen depending on the needs of the display and that is the head or the person of the model that is displayed. The remaining parts of them stand out communicative elements in real life at the moment of displaying the details of the passive elements of the complexity of the display. In all mentioned modes of communication it is possible to talk between a large number of autonomous virtual 3D models of avarats, real models of people and their clones.

3 Description

The general goal of photorealistic rendering systems is to be able to generate 3D photorealistic model or render clones of real-life models of people, animate them thereby, and present and behave as real-persons in a virtual 3D world. who are surrounded. The virtual world may represent and contain another virtual 3D avatar model that is autonomous within it and may maintain dialogue and other communication between another autonomous 3D avatar model or clone of a real human model. In combination, the automated virtual 3D avatar model understands expressions and speech from the real person (transmitted and presented by its 3D clone) and can establish communication using a variety of verbal and non-verbal signals. The dialogue achieved in this way is simulated in a 3D virtual environment in which it can be present from different locations anywhere in the world using an Internet connection. Using such a system requires solving many different situations in many empowerments such as pre-processing and real-time rendering of a large number of images, video-audio processing, networking of system users, use of vegetative intelligence, pre-processing virtual reality and defined 3D animations. Also, when designing combined systems, they are shared with all users, with each module representing a separate logical unit from the entire system in which it can be present. The modules can be divided into a number of different groups, but the most represented and exploited are:

- 3D face modeling and reconstruction,
- facial animation,
- expression recognition,
- audio and video processing,
- interpretation and response generation and
- visual synchronization.

Virtual 3D models of face avatars can be represented by generic modeling or by cloning using two orthogonal photographs, which in practice are usually taken in side and front views. This process is also called model adaptation (modeling) as it involves transforming the generic face model into a specific pre-modeled and customized virtual 3D face from a real human model. In order to define and characterize the shape and morphology of the model-face for each view, it is first necessary to prepare 2 (two) 2D templates that contain the characteristic points of the generic modeled view of the face-model. In cases of rendering, if the two views of the face to be animated-representations have different heights and sizes, it is first necessary to carry out a normalization process and then to match the 2D templates with the corresponding characteristics of the reference images. Also, this technique uses specially structured predefined templates of the elements that are displayed, such as the contours of the hair, but also as filtering methods for other features that define the 3D model itself, such as the eyes, ears, chin and nose. By combining two separate sets of 2D coordinates, a display is performed when calculating the 3D coordinates during display and thus provides a set of target positions of the characteristic points that are displayed on the 3D model itself. In cases of pre-defined critical points that are generic and do not contain any characteristics, they are modified and deformed by a process called - Dirichlet Free Form Deformation (DFFD). DFFD is a commonly used free form deformation (FFD) method that combines traditional FFD with various other interpolation methods of different deformation methods that are based on different diagrams. Also a very significant element is that the DFFD does not define any heat limitations when controlling the control grid itself (Figure 6) and the control points can be defined by free choice in the environment itself that is used. Once the FFD setup is complete, one can move on to defining the model using a set of feature points that are defined on the surface of the generically modeled 3D face as DFFD control points. In order to get the best photorealistic rendering possible, texture mapping can be used to show the smallest details such as facial features in many cases not generated when modeling the 3D virtual model. The following image – Figure 7 illustrates the different steps involved in modeling and rendering a virtual 3D model of a real human face.

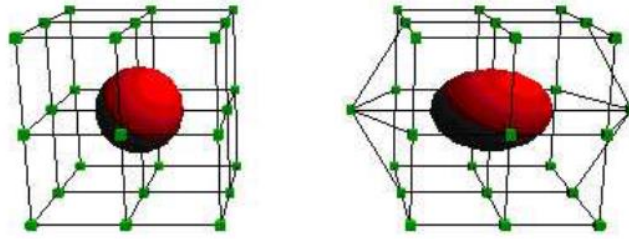


Figure 6

For a realistic rendering, we use texture mapping to reproduce the small details of facial features that may not appear in the gross montage of the model. Figure 7 shows the different steps to construct a 3D face model for the clone.

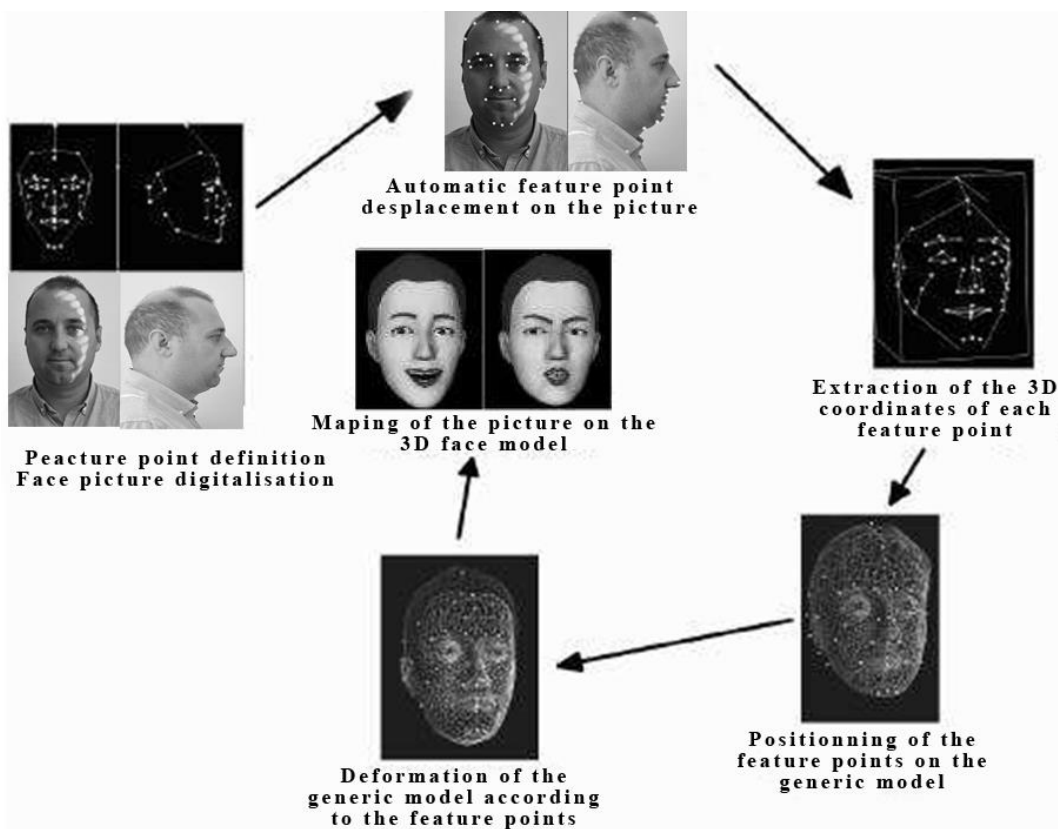


Figure 7

3D virtual models face and is an irregular defined structure polygon mesh. The 3D model of a virtual face is divided into polygons on which the muscle activities are expressed depending on the part, and depending on their application, a proportional free-form deformation of the model itself is simulated. By transforming the generated 3D model to a license-free base structure the new 3D model can be animated and its animation can be controlled with several levels of control. The lowest level of control of a 3D model contains a minimum of 65 minimum actions of (MPA) that are associated with defined and associated with the style of each muscle used in rendering the color 3D form. Each MPA represents a separate polygonal part that simulates certain numerical parameters for the movement of the model with which certain parts of

the face are controlled such as changes or blinking of the eyes. By using a minimum of 65 detected features of MPA it is possible to present and display a limited number of expressions on the virtual model of the 3D face and express a limited number of phonemes separately. By using, applying and determining the number of MPAs, phonemes and expressions are used on 3D virtual faces, while using the highest level you can control the animations in the content of the speech and express the expression by defining their duration and synchronizing them with the speech. Depending on the use of 3D animation on the model and the type of application, different levels of animation control are used. These scenarios are illustrated in the following figure - Figure 8.

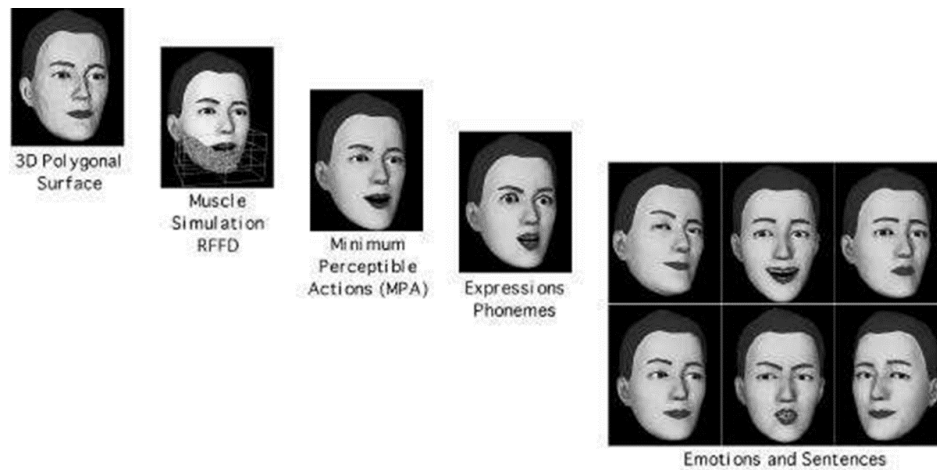


Figure 8

Real-time recognition of different expressions from real or virtual 3D models of face avatars is a laborious and complex process of rendering animations. The way animations are displayed is seen when rendering – the presentation of the model needs to be done early on. In order to recognize and express a certain facial expression to a photo-realistic degree and show it with highly accurate animation in real time, it is a painstaking and precisely defined work of the 3D model itself, and as an inevitable part of all of that, there are simplifications that should be shown in relation to the rendering itself is alive most of the time we are focused on certain few arms of the face itself which are used in tracking the 3D model itself which is being animated. As the main feature of this display and animation of a virtual 3D face, it can be written that it contains a selection of points that are defined and adjusted to the image of the model being displayed. Reporting custom animation and that initialization "mask" which is defined and represents a set of certain defined animation reviews can be interactively adapted to any image or real model while providing details of measurements and definitions of characteristic features of the character being displayed. Also an important feature of this display method is the representation and storage of information such as skin color, eye color, hair, etc. This method can be of certain certain characterists are based on the identification of samples that contain degrees of color from different parts of the face itself and the detection of the characteristics of the points. Also, the features of the points that are used and extracted to recognize certain facial expressions are mostly concentrated on the shape of the eyebrows, the shape of the eyes, the size of the face, the size of the face and the contours of the hair of the model itself. Data that is sampled from something like a predefined framework for animating a 3D model can also be used in functions that are relatively easy to follow (for example from the throat) thus experiencing accumulations of rendering errors. In order to display and reproduce features of 3D people, separate individual feature mappings from MPA features to standard motion parameters should be performed when animating 3D face models. Such mapping allows us to photo-realistically imitate the facial expression from the real model to its 3D photo-realistic model. This example is shown and illustrated in the following figure - Figure 9 and is an illustration of how to animate a 3D virtual face from a clone using captured facial expressions from a real model.

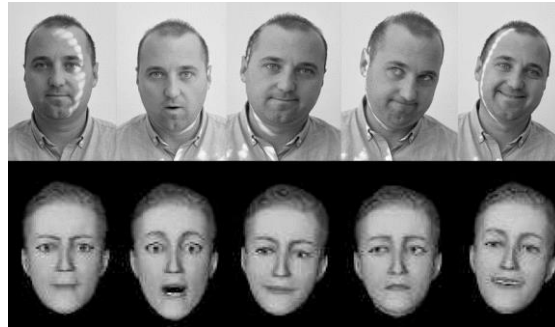


Figure 9:

All this information still doesn't tell us enough about the actual person's mood and/or emotion. A person's mood and emotions are important information to be fed into the process of formulating an autonomous virtual human's response. We use a rule-based approach to infer a person's emotions. These rules are simply mapping active MPAs to a given emotion, however, they are limited to only a few types of emotions. Another approach to basic emotion recognition is to use a neural network with automatically extracted facial features as input associated with each emotion (surprise, disgust, happiness, fear, anger, and sadness). Using a neural network approach where the input is extracted data from the video and the output is one of the six emotions, the difficulty remains to identify a set of underlying emotions. This can be partially addressed by identifying the dominant features of the underlying emotion displayed and masking the others.

4 Using and processing audio

In the audio signal processing process, most of the dialog content in its processing is the audio signal processing, and it represents many meanings in the process itself. From previous testing in a limited contextual vocabulary of expression that is dependent on the speaker on the opposite side, it can be concluded that the success rate of the processing itself can reach up to 90% without any particularly large and significant efforts. Today in reality there are many speech recognition techniques used by animators themselves when expressing themselves in the virtual world and they involve many processing processes, the most significant of which are: digital recognition and processing of real speech samples and processing of signals that are acoustic (it is generally performed in the analysis of different participants in the speech, in the recognition of phonemes, in the derivation of groups of phonemes that are then expressed in words, in the application of a Markov modeling system, which is currently a semi-polar method of recognition and basic syntactic recognition in spoken language that can help in the process of recognition itself). The operation and application of these techniques are time-consuming and may not be suitable for use in other real-time applications. It can also be noted that when using a simplified context, real-time recognition is possible. Before formulating the answer to be obtained, the semantics of the input speech is essential in the application of the audio analysis itself, and analogously, the text in the speech or its synthesis represents a very important part of the audio processing itself. In processing, a common procedure is to divide textual speech into small units that generally represent phonemes and represent the smallest meaningful linguistic unit that contains the corresponding audio signal. Since it is not at all an easy process to combine them to obtain quality speech, the most commonly used solution is the use of diphones instead of phonemes that contain transitions between pairs of phonemes. This procedure significantly increases the number of elements that need to be processed in the database itself, and therefore the database itself, but the quality of the resulting speech is much better and higher quality. In order to obtain a voice that is more human-like, certain features (inflections) are added that correspond to the characteristic punctuation used. Also, in order to quality-animate 3D photorealistic avatars and their face using an audio input, phonemes are mapped to their corresponding visual output which can be called a pendent. Since the image itself is a set of MPAs, it can be applied to any 3D virtual photorealistic avatar and its face. The following figure – Figure 10 illustrates the display between two different 3D virtual avatar models.

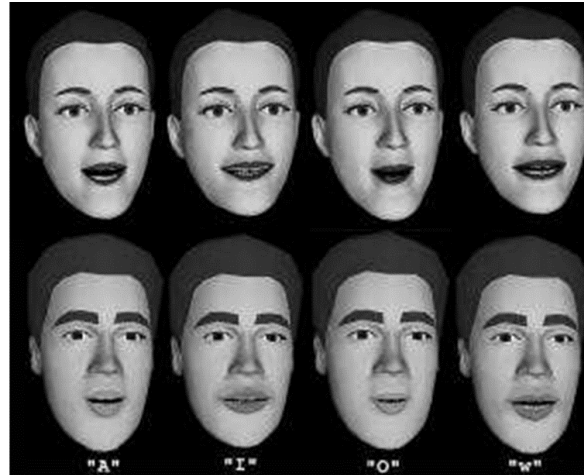


Figure 10

5 Interpretation and response generation

In order to achieve a dialogue between a 3D model clone of a real model and a 3D virtual avatar that is autonomous, the autonomous 3D avatar model should be able to "understand" the speech and the signs they express expressed by the 3D clone model. Such realization of dialogue requires the use of a function-module which is intelligent to act as the brain of the autonomous 3D virtual avatar of the student in the dialogue being realized. Information that is processed by the speech text from the 3D model of the clone from the real model and such data that they express are defined by the expressions not the speaking person. Also, this analysis technique provides possibilities and can be extended when combined with other real voice processing techniques. This situation is illustrated in the image that follows - Figure 11.

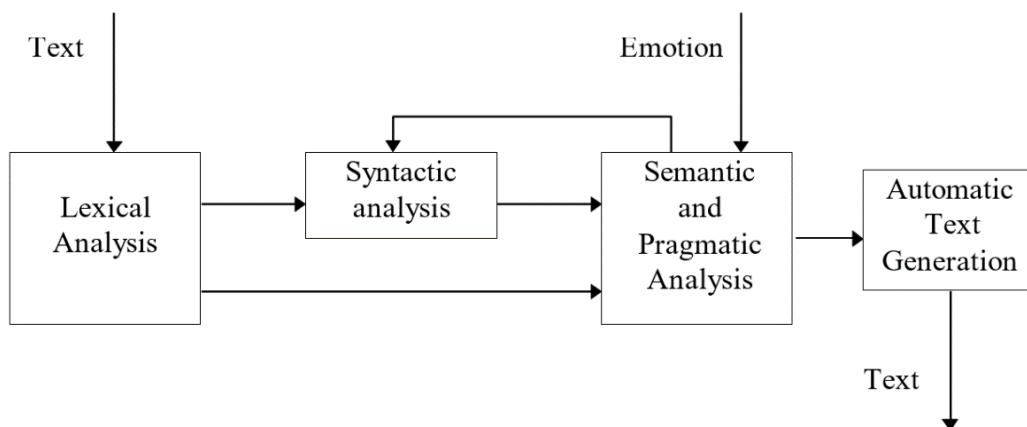


Figure 11

For each word uttered by the 3D virtual avatar, the algorithm analyzes the information previously stored in the lexicon and then the syntactic analysis provides a speech based on grammatical rules analyzing each sentence in order to sense the relationships among the different groups of words used in defining the context of pronunciation of different groups of words. Also, the own semantic analysis of the words commented by the virtual 3D avatar gives the real-true meaning during a specific dialogue and thus generates the output used in the generation of automatic responses. Emotions that are transferred from a 3D photorealistic model of a clone to be essentially expressed by applying such an analysis technique of which the reports can be familiar ones that are found in the real world. In order to obtain a realistic appearance of the 3D

photorealistic avatar while at the same time a person displays different expressions in the audio and 3D visual appearance of the virtual avatar must be synchronized when displaying in interaction. Also in order to photorealistic display of photorealistic 3D avatars system and algorithms used for dialog should share how communication between clients and server architecture is used in the defined virtual environment. Participating in one type of virtual environment has the real one represented by the clone's own 3D photorealistic model and multiple other external 3D photorealistic models of those that can be on or off while completing the same dialogue. Also included in the virtual ones are the various deformations of the elements that are an integral part of the 3D virtual environment itself, the expression of the speech synthesis parameters of the 3D virtual avatar that represents an autonomous real character and the speech that is compressed to the clients or 3D photorealistic models. from clones. The following figure - Figure 12 illustrates such an interaction between a 3D photorealistic virtual model and another 3D photorealistic virtual model of real human models while racing in a 3D virtual scene in which a third is racing. are only observers of certain scenes that are shown.



Figure 12

The figure that follows – Figure 13 illustrates to us a complete overview of the events that show the movement of information between the various modules for rendering 3D photorealistic models of virtual avatars. In mutual cases they enter to be taken from different forms of modes they use for audio and video inputs that should be other inputs such as 3D markers that have gestures and body positions.

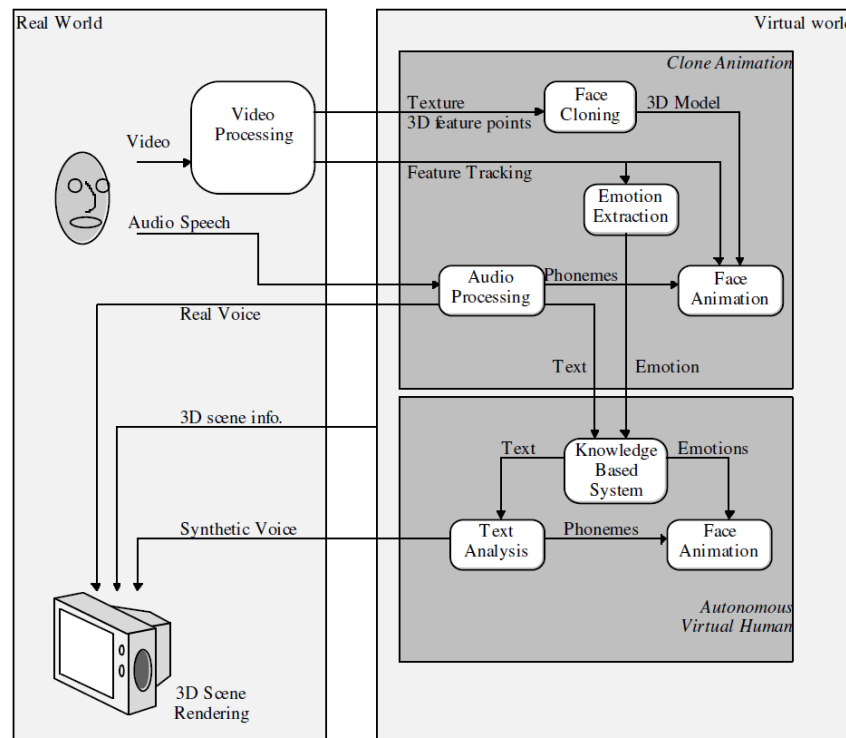


Figure 13

Today, if we want to illustrate a 3D photorealistic presentation before it illustrates any speech-text, first the audio signal is analyzed and the phonemes are expressed in the text that follows. In order to synchronize it with its own animation that is performed, it is necessary to define the duration of each illustrated phoneme. Also to display visual information about the photorealistic virtual 3D avatar that is displayed while all the parts and its 3D features that are used in the modeling of the virtual face itself to the 3D model represents which clone from a real model and features that are needed when passing the only animation is needed to perform preprocessing and define certain displays. As a separate and independent entity, the autonomous virtual 3D avatar or person using different technologies is able to receive the data-emotions that are expressed at the very input or beginning of the algorithm. Then this information is processed with different algorithms and used in different modules that are interested in knowledge but also in the future they may be able to come up with an answer that would be logical without obstacles and at the same time be able to express and displays text. Text processing retroactively generates phonemes that are later applied when performing facial deformations as well as simulating synthetic speech that simulates the current emotion. The following silica – Figure 13 shows thus and illustrates the interaction and dialogue found between a real human model and a 3D virtual model of a face but also provides opportunities for dialogue and interaction between clones to be extended with multiple 3D photorealistic models or clones of real models. Also, the example shown in figure 13 shows that it is indeed possible in a limited context of the body to feel and see different families of own 3D models or facial expressions from real models, but also to recognize and perform those real expressions and such on models of faces and thus to discover a virtual dialogue in real time between the different forms of presentation of the models. Also, such combinations of displaying the expression generated by the own application can be mapped to a virtual avatar that represents a real model and is able to simulate the movement and speech of a real person-model and to a 3D photorealistic avatar that can is autonomous and is able to have and talk with a real person - the 3D model that is represented by his clone, and thus expresses and simulates certain signs when they realize a dialogue.

6 Standardization

In order to set, determine certain parameters during animation and set some order today MPEG-4 standard is defined which offers the possibility to write code for those 3D models of virtual avatars relatively easily and is not able to synchronize and perform transmission of those pairs used to define other animations that are specific to a particular 3D virtual avatar or its model and a subset of MPEG-4 today is the most widespread and more widely used SNHC (Synthetic Natural Hybrid Coding). Combined complexes to animate a 3D photorealistic model of a face that is defined by a set of face defining parameters (FDP) and use simple parameters for it and are able to encode the shape and texture of the 3D model itself. It also contains information and is able to express itself, different expressions and engage in dialogue. In reality the FAP contains data on the expressions that make up the face model with which they are associated and together with what relates to the face. The expressions or actions that the 3D models themselves reveal are a complex set of basic actions that the 3D models themselves perform and enable the representation and rendering of photorealistic and natural expressions of the face itself. In order to get the right lips, an example should be given, as defined as it depends on them, and further during animation, the internal/external contours shown by the 3D face model itself are taken into account. Also over-defined FAP values for certain elements can lead to actions that are not normally possible to perform by real human models, therefore it is best to define FAPs according to the need and purpose of the 3D model that will use it (for example different sets of FAPs are needed to animate 3D models in children's cartoons and games such as FIFA,NBA). For all the parameters involved in a particular movement are expressed in terms of the parameters defining the facial animation (FAPu). Also match the elements and the distances between the most important features of the in-place model (for example the distance between the space and the nose or the distance between the eyes). By adding a number of critical points, complexities are obtained and costs are required. A parameter group can contain up to three high-level parameters. In order to fully x-ray and display the visages, it should be possible to enable their direct rendering of the 3D face model itself in real time, which is displayed without the influence of other parameters in order to realistically display and improve the users themselves by using the third part. parameters..

After the displayed and illustrated parameters of the data that are listed are not certain characteristics for use in full and definition of all parameters, the minimum number of parameters that should be used should be defined. For example, the previously mentioned parameters used to display facial expressions should allow the definition of facial expressions at a high level. Also in reality quite often FDP is expressed to adapt certain 3D face models to other 3D face models that contain features. They can specify features such as the shape of the eyes, the size of the color, the shape and beauty of the eyebrows, a 3D polygonal model with those coordinates used for the texture- if it is not important mapped and others. The following image – Figure 14 illustrates and displays some of the more important critical points and parameters used today in 3D facial animation of avatars.

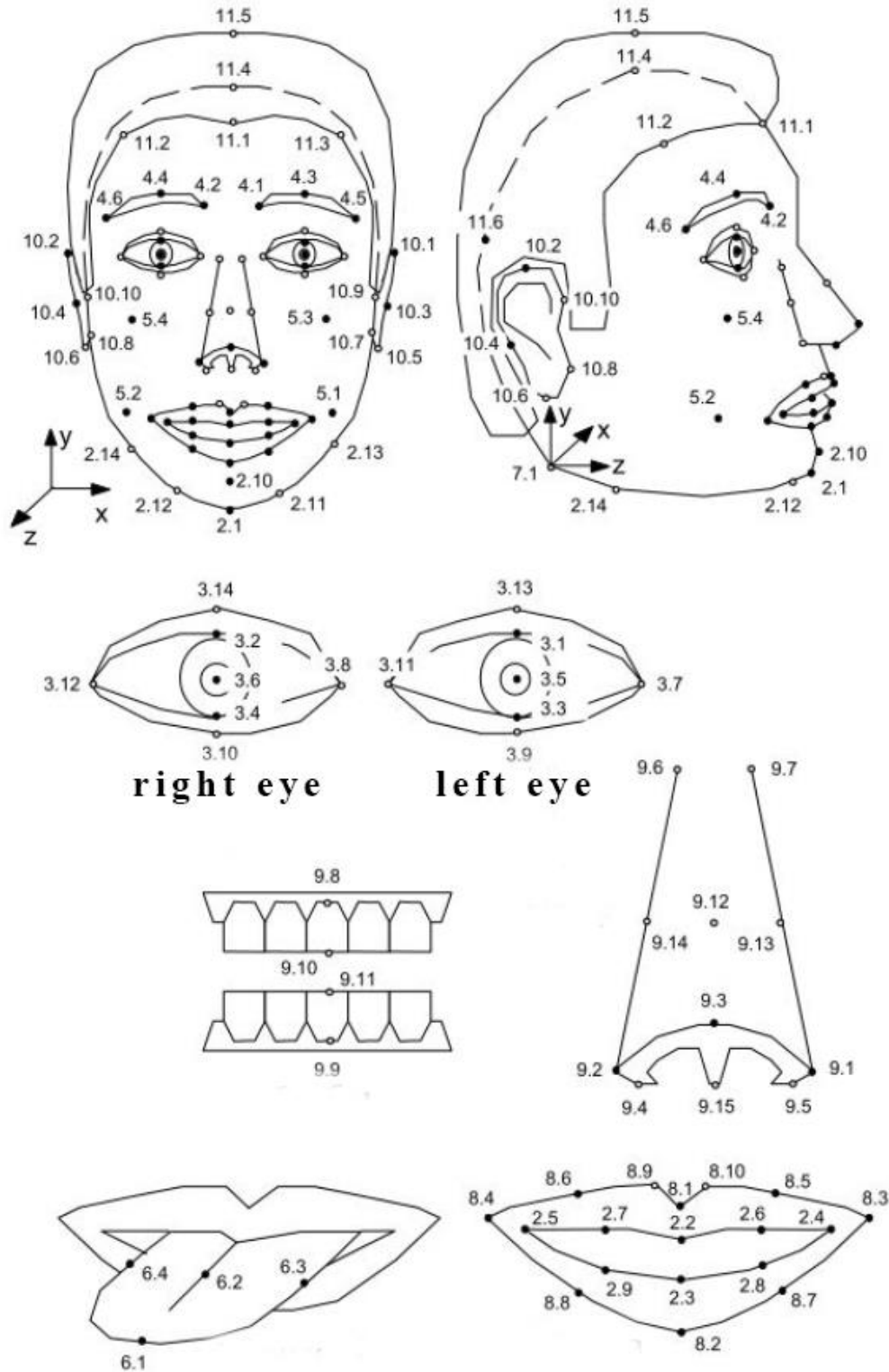


Figure 14

7 Conclusion

Today, with the help of a computer, the communication of a certain 3D model of a photorealistic avatar-Bot, which is displayed on a 2D or 3D output device with a real model/character/person, has already become a reality. The same without any specific programs and requirements for animators who have previously prepared and trained are able and can possess enough lids to achieve effective interaction and dialogue between realistic people who are in real worlds and believers of light 3D avatars who meet in a virtual world but also able to offer a new connection and interaction and connection between machines and humanity. This paper describes the interaction and communication between real person models that can be represented by their 3D virtual clones or cloned virtual 3D faces and 3D virtual avatars or faces. The systems that are used today are not able to provide insight into different solutions that can be used in the reconstruction of virtual 3D avatars, their faces or their genders that have the ability to use and receive the different shapes and other models of real - real persons without their physical attention. They are also able and have capabilities during the occurrence of their own real-time interaction to include different types of analysis (for example syntactic or semantic analysis) of the requests transmitted by a 3D model of a photorealistic avatar or its clone is expressed of its own facial expressions, character or speech which later in the course of interaction is used to display a response that is believable on the part of a virtual 3D avatar or its virtual character. such realized communication or dialogue may contain verbal or non-verbal forms of the 3D model or the person parallel to the synchronization of the voice itself – the audio used. With all the capture of the visual and sound information and obtaining a real presentation of a certain 3D virtual model or a virtual 3D person in the base of the same, a learner should possess a number of information about his perception and intelligence that owns the real model. The possession of this knowledge can be used and expanded when using different applications, when using different interfaces that can be intelligent or virtual in themselves, with which different levels of communication are achieved, for example: between a real character-human and machine-3D virtual avatar, realization of teaching and accompaniment in virtual schools and eyes in virtual schools, realization of virtual teaching, performance of virtual work, etc. Such realized communications provide opportunities to imagine and observe a situation or a situation that can be real invented for a specific case a 3D model of an avatar or its clone where a real person in a virtual environment is populated and other 3D virtual avatars are represented or their clones who have autonomous knowledge in that virtual world. But in reality and from today's point of view, although this technology is very advanced and is constantly advancing due to the complex and sensitive nature of this technology that uses it today, it can be accurately taken that it is possible to check the factors more in order to achieve communication and dialogue in real time with an interaction that would be believable in real life. In the future, the recognition, application and integration of other bodily gestures, the realization of interaction and communication that will take place is expected to move to a higher level that has not been explored. Comments go in the direction of communication between virtual people using facial gestures and other real models or people. In reality, 3D modeling of photorealistic avatars or cloning of virtual people as a whole here includes and animation of such virtual 3D avatars requires a lot of research. The following image – Figure 14 illustrates and depicts a situation in which virtual 3D human models are communicating and interacting with a known person who is also virtual.



Figure 15

References

- [1] Thorisson Kristinn R (1997), “Layered modular action control for communicative humanoids”, Proc. Computer Animation ‘97, IEEE Computer Society, pp. 134-143.
- [2] Lee WS, Kalra P, Magnenat Thalmann N (1997), “Model based face reconstruction for animation”, Proc. Multimedia Modeling (MMM)’97, Singapore (to appear).
- [3] Escher M, Magnenat Thalmann N (1997), “Automatic cloning and real-time animation of a human face”, Proc. Computer Animation ‘97, IEEE Computer Society, pp. 58-66.
- [4] Kalra P, Mangili A, Magnenat Thalmann N, Thalmann D (1992), “Simulation of facial muscle actions based on rational free form deformations”, Proc. Eurographics ‘92, pp. 59-69
- [5] Magnenat Thalmann N, Kalra P, Pandzic IS (1995), “Direct face-to-face communication between real and virtual humans”, International Journal of Information Technology, Vol. 1, No. 2, pp. 145-157.
- [6] Philips Speech Processing, URL address <http://www.speech.be.philips.com>.
- [7] Pure Speech, URL address <http://www.speech.com>.
- [8] Vocalis, URL address <http://www.vocalis.com>.
- [9] Doenges Peter, Lavagetto Fabio, Ostermann, Joern, Pandzic Igor Sunday, Petajan Eric, “MPEG-4: Audio/video and synthetic graphics/audio for real-time, Interactive Media Delivery”, Image Communications Journal, Vol. 5, No. 4, May 1997.
- [10] D. DeCarlo, D. Metaxas, ‘Optical Flow Constraints on Deformable Models with Applications to Face Tracking’, CIS technical report MS-CIS-97-23.
- [11] E. Cosatto, H.P. Graf, ‘Sample-Based Synthesis of Photo-Realistic Talking Heads’, Computer Animation 1998, pp103 – 110
- [12] C. Kouadio, P. Poulin, P. Lachapelle, ‘Real-Time Facial Animation based upon a Bank of 3D Facial Expressions’, CA98, 1999, pp128 – 136
- [13] SNHC, ‘Text for ISO/IEC FDIS 14496-1 Systems (2nd draft)’, ISO/IEC JTC1/SC29/WG11 N2501, 1997. ISO/IEC JTC1/SC29/WG11 N2501, Nov., 1998
- [14] G. Sannier, S. Balcisoy, N. Magnenat-Thalmann, D. Thalmann, ‘VHD: A System for Directing Real-Time Virtual Actors’, The Visual Computer, Springer, 1999.
- [15] W. S. Lee, M. Escher, G. Sannier, N. Magnenat-Thalmann, ‘MPEG-4 Compatible Faces from Orthogonal Photos’, CA 99, 1999 pp186-194.
- [16] AIML webpage. <http://alice.sunlitsurf.com/>, November 2002.
- [17] Michael Johns and Barry G. Silverman. How emotions and personality effect the utility of alternative decisions: a terrorist target selection case study. In Tenth Conference On Computer Generated Forces and Behavioral Representation, May 2001.
- [18] S. Kshirsagar, S. Garchery, and N. Magnenat-Thalmann. Deformable Avatars, chapter Feature Point Based Mesh Deformation Applied to MPEG-4 Facial Animation, pages 33–43. Kluwer Academic Publishers, July 2001.
- [19] S. Kshirsagar and N. Magnenat-Thalmann. Amultilayer personality model. In Proceedings of the second International Symposium on Smart Graphics, pages 107– 115. ACM Press, June 2002.

Mathematical modelling of scalability and performance testing methods for evaluating server capabilities

Danail Runchevski † . Atanas Hristov †. Agim Rushiti ‡

Email(s): danail.runchevski@cns.uist.edu.mk
atanas.hristov@uist.edu.mk
agim_rushiti@yahoo.com

† University for Information Science and Technology “St. Paul the Apostle”, Ohrid, Macedonia

‡ Ministry of Science and Education, Skopje, North Macedonia

Abstract

In the era of rapidly evolving technological landscapes and growing user demands, the scalability and performance of servers play a critical role in ensuring efficient and reliable service delivery. This research paper explores the various testing methodologies designed to evaluate the capabilities of servers concerning scalability and performance. It delves into the intricacies of testing methodologies, ranging from stress testing and load testing to performance profiling and benchmarking. The paper highlights the significance of these methodologies in assessing server response under varying workloads and traffic conditions, thereby enabling administrators and organizations to identify potential bottlenecks and optimize server configurations for enhanced performance. Furthermore, the research scrutinizes the challenges and complexities associated with accurately simulating real-world scenarios during testing, emphasizing the importance of comprehensive test environments that reflect the actual usage patterns and demands on servers. By presenting a comprehensive analysis of scalability and performance testing methods, this paper aims to provide valuable insights into the best practices for evaluating and enhancing the capabilities of servers in diverse operational environments.

Keywords

Performance, testing, scalability, server.

1 Introduction

Amid the contemporary digital landscape, the efficiency and adaptability of server systems are pivotal for the seamless delivery of services and applications. The escalating demands imposed by evolving workloads and fluctuating user traffic have underscored the criticality of gauging the scalability and performance capabilities of servers. In response, the adoption of rigorous testing methodologies has become indispensable in evaluating how servers manage increased loads and sustain consistent performance levels. By subjecting servers to various stress tests, load simulations, and performance evaluations, IT professionals can glean invaluable insights into the operational behaviour and constraints of their server infrastructure, guiding strategic decisions related to resource allocation and infrastructure augmentation. This research endeavours to provide a comprehensive exploration of the diverse testing techniques employed to assess server capabilities within multifaceted operational environments.

It seeks to offer a detailed analysis of stress testing, load testing, performance profiling, and benchmarking, emphasizing their pivotal role in identifying potential bottlenecks and ensuring uninterrupted operations, particularly during periods of heightened demand. The research will scrutinize the complexities associated with creating test environments that faithfully emulate real-world conditions, underscoring the significance of designing scenarios that mirror authentic usage patterns and traffic dynamics. Through an in-depth examination of scalability and performance testing, this study aims to contribute to the establishment of best practices for optimizing server systems, equipping organizations to meet the escalating demands of the digital sphere while upholding peak performance and dependability.

2 Scalability

In the dynamic landscape of digital services and applications, the ability of server systems to scale efficiently in response to increasing workloads and user demands is crucial for maintaining optimal performance and uninterrupted service delivery. The scalability of a server refers to its capability to accommodate growing data volumes and user traffic without compromising response times or system stability. This section delves into the fundamental aspects of scalability and explores the key considerations for achieving scalable server architectures.

A. Understanding Scalability Types

Scalability can be categorized into different types, including vertical scalability and horizontal scalability. Vertical scalability, also known as scaling up, involves enhancing server capabilities by adding more resources, such as CPU power, memory, or storage, to a single server.

Horizontal scalability also referred to as scaling out, focuses on distributing the workload across multiple servers in a network, allowing for the seamless addition of new servers to handle increased traffic and data processing requirements.

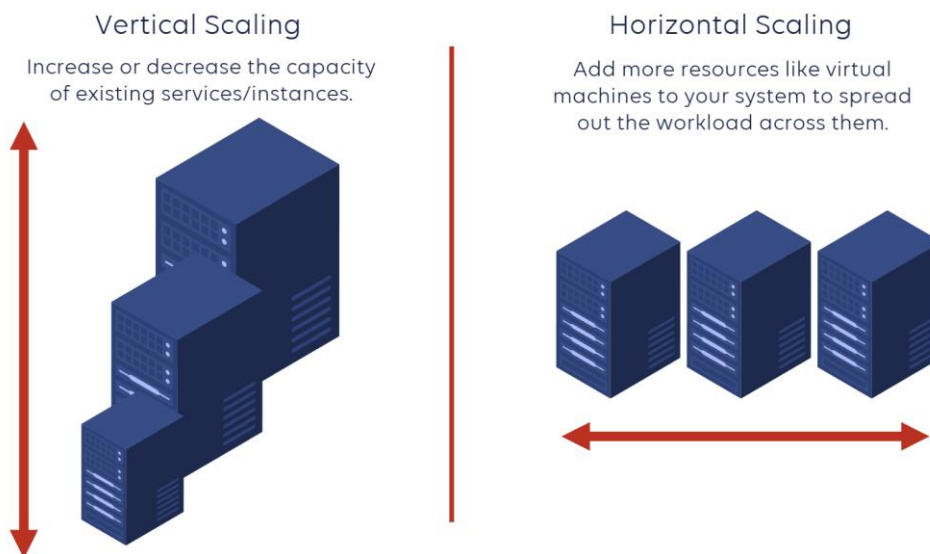


Figure 1: Vertical and horizontal scaling

a. *Vertical Scaling (Scaling Up):*

Vertical scaling, also known as scaling up, involves enhancing the capacity of a single server by increasing its processing power, memory, or storage resources. This approach allows organizations to improve the performance of a server by upgrading its hardware components. By adding more powerful CPUs, increasing the RAM capacity, or integrating faster storage devices, organizations can boost the processing capabilities of the server, enabling it to handle more significant workloads and complex tasks. Vertical scaling is typically suitable for applications that require substantial computational power, such as high-performance computing, database management, and resource-intensive data processing. While vertical scaling can provide immediate performance improvements, it may pose limitations in terms of cost-effectiveness and the maximum threshold of resource expansion, making it essential for organizations to consider the scalability limits of their hardware infrastructure carefully.

b. *Horizontal Scaling (Scaling Out):*

Horizontal scaling, also referred to as scaling out, involves expanding server capabilities by adding more servers to the existing network infrastructure. This approach allows organizations to distribute the workload across multiple servers, thereby improving overall system performance and accommodating growing user demands. By implementing load-balancing mechanisms and distributed computing architectures, organizations can ensure that incoming requests and data processing tasks are evenly distributed across the network of interconnected servers. Horizontal scaling is particularly advantageous for applications that experience a high volume of user traffic, such as web servers, content delivery networks (CDNs), and distributed computing platforms. While horizontal scaling offers enhanced scalability and fault tolerance, it may introduce complexities related to data synchronization, network communication, and system management, necessitating the adoption of robust network architectures and scalable data storage solutions to ensure seamless operations and data consistency across the distributed server environment.

B. Scalability Challenges and Solutions

Despite the benefits of scalability, organizations often encounter various challenges when implementing scalable server architectures. Common challenges include maintaining data consistency across multiple servers, managing inter-server communication, and ensuring seamless load distribution without compromising data integrity. Implementing effective solutions such as data partitioning, load balancing algorithms, and distributed caching mechanisms can help mitigate these challenges and facilitate the smooth expansion of server capacities to meet evolving business demands.

C. Cloud-Based Scalability

Cloud computing has revolutionized the scalability paradigm by offering flexible and on-demand resources that can be dynamically scaled based on fluctuating workloads. The adoption of cloud-based scalability solutions, such as auto-scaling, elastic load balancing, and serverless computing, enables organisations to efficiently manage sudden surges in traffic and optimize resource utilization while minimizing operational costs. Exploring the benefits and challenges associated with cloud-based scalability is crucial for organizations aiming to leverage the scalability advantages offered by cloud service providers.

3 Testing types

To comprehensively evaluate the capabilities of server systems, a strategic and multifaceted testing approach is crucial. This section provides an in-depth exploration of key testing methodologies that are commonly employed to assess server scalability and performance.

A. Stress Testing

Stress testing constitutes a fundamental aspect of evaluating server capabilities, as it involves pushing the server to its limits by subjecting it to extreme workloads and traffic levels. This testing method helps administrators identify the maximum operational capacity of the server, determining the thresholds beyond which the system might become unstable or fail. By deliberately imposing excessive loads and monitoring the server's response, professionals can gain critical insights into its resilience and potential vulnerabilities under highly demanding conditions. Stress testing enables organizations to ascertain the robustness and reliability of their server infrastructure, facilitating proactive measures to reinforce system stability and prevent potential disruptions during peak usage periods.

B. Load Testing

Load testing plays a pivotal role in assessing how a server performs under anticipated or expected workloads, simulating realistic user traffic and application usage patterns. By generating controlled loads that mimic actual user behaviour, administrators can evaluate the server's ability to handle concurrent user requests and transactions without compromising performance or response times.

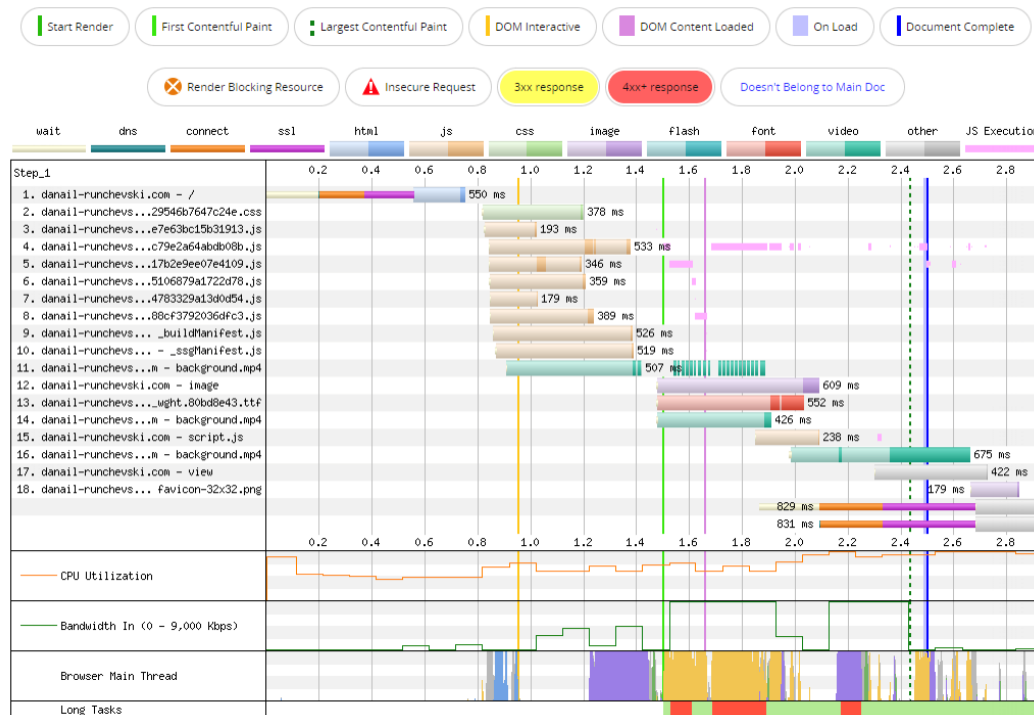


Figure 2: Loading time testing graph example

Load testing enables organizations to identify performance thresholds, measure response times, and pinpoint potential bottlenecks that might arise under specific load conditions. By conducting comprehensive load tests, professionals can make informed decisions regarding resource allocation, capacity planning, and infrastructure scaling, ensuring that the server can seamlessly accommodate increasing user demands and fluctuations in traffic volume.

C. Performance Profiling

Performance profiling involves the meticulous analysis and monitoring of the server's behaviour under specific workloads, aiming to identify performance bottlenecks and areas for optimization. This testing methodology facilitates a comprehensive assessment of resource utilization, latency patterns, and potential inefficiencies within the server infrastructure. By closely monitoring various performance metrics, such as CPU usage, memory consumption, and network throughput, administrators can pinpoint critical areas that require optimization and fine-tuning. Performance profiling enables organizations to implement targeted improvements, optimize resource allocation, and enhance overall system responsiveness, thereby ensuring a consistently high level of performance and user satisfaction.

D. Benchmarking

Benchmarking serves as a crucial tool for comparing the performance of a server against predefined metrics or industry standards, facilitating an objective assessment of its relative efficiency and effectiveness. By conducting standardized tests and measurements, administrators can gain a comprehensive understanding of the server's strengths and weaknesses, enabling informed decisions regarding hardware upgrades, configuration adjustments, and optimization strategies.

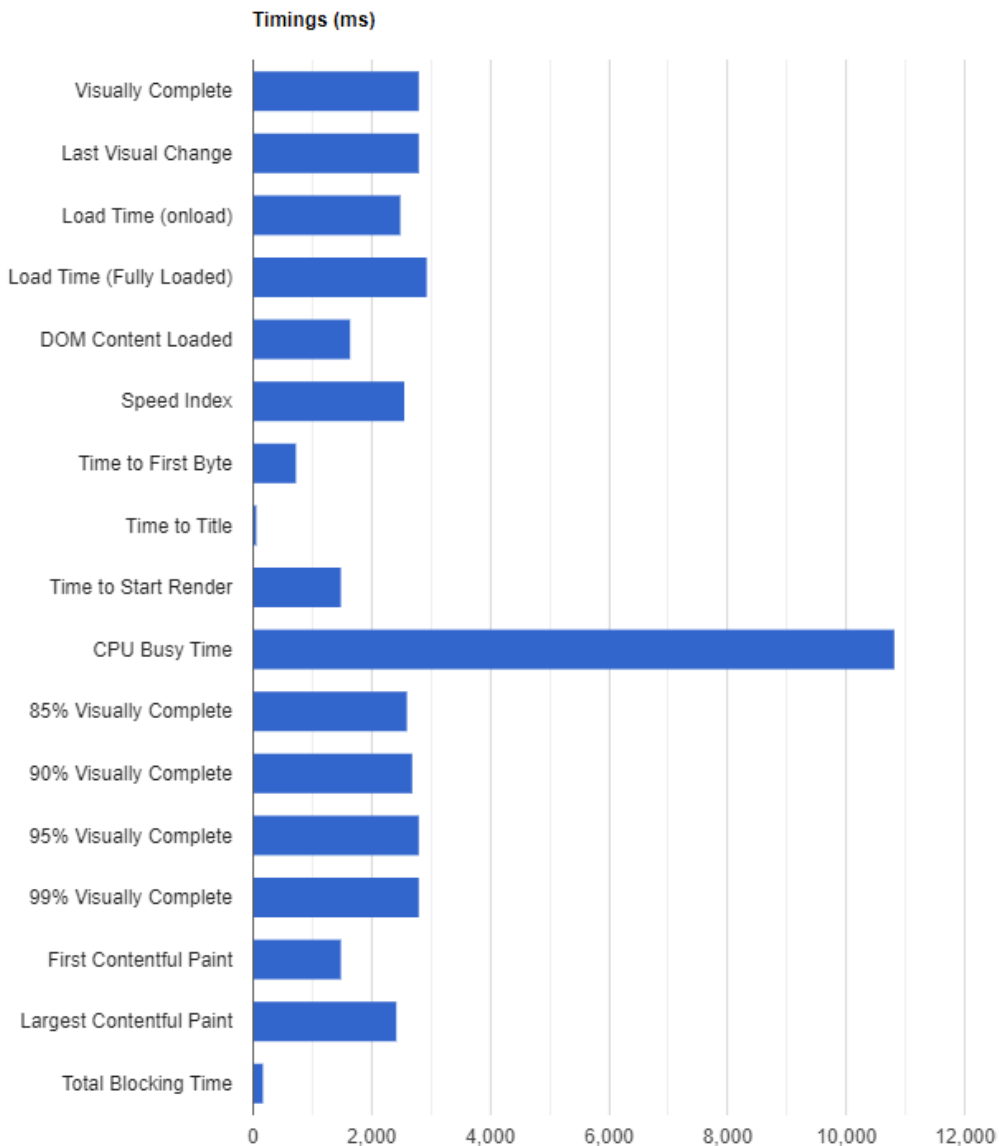


Figure 3: Responsive filmstrip

Benchmarking provides valuable insights into the server's processing capabilities, data transfer speeds, and overall responsiveness, allowing organizations to identify areas for improvement and implement tailored solutions to enhance system performance and reliability.

4 Evaluation

Effectively assessing the efficacy of server scalability and performance testing methodologies necessitates a comprehensive evaluation process that considers various critical factors and performance indicators. The following section outlines the key considerations and criteria for evaluating the effectiveness of scalability and performance testing methodologies in ensuring optimal server performance and seamless service delivery.

A. Comprehensive Scalability Analysis

Evaluating the effectiveness of scalability testing methodologies involves assessing the efficient utilization

and allocation of server resources during peak workloads. The evaluation process should focus on identifying the extent to which testing methodologies facilitate optimal resource distribution and enable the server to accommodate fluctuating workloads without compromising performance or stability. Assessing the scalability testing methodologies should include a thorough examination of the scalability limits and thresholds of the server infrastructure. The evaluation process should aim to determine the maximum capacity at which the server can efficiently handle incoming workloads and user traffic while maintaining consistent performance levels and ensuring seamless service delivery.

B. Performance Testing Efficacy

Evaluating the efficacy of performance testing methodologies requires a detailed analysis of the server's response times and throughput under varying workloads. The evaluation process should focus on measuring the average response times, transaction processing rates, and data transfer speeds to assess the system's responsiveness and its ability to handle concurrent user requests without delays or performance degradation. Assessing the performance testing methodologies should include an in-depth examination of the error rates and system reliability during stress testing and load testing scenarios. The evaluation process should aim to identify the frequency of errors, system failures, and performance bottlenecks, enabling administrators to implement proactive measures to enhance system reliability and minimize potential disruptions during peak usage periods.

C. Future Testing Methodologies Appraisal

Evaluating the future testing methodologies should focus on assessing the adaptability of AI-driven testing frameworks and DevOps integration to dynamic workloads and evolving business requirements. The evaluation process should aim to determine the extent to which these testing methodologies enable organizations to proactively address scalability challenges, optimize system performance, and deliver seamless and reliable services to end-users in rapidly changing digital environments. Assessing the future testing methodologies should include an examination of the capabilities of AI-driven testing frameworks and DevOps integration in conducting predictive analysis and facilitating proactive resolution of performance bottlenecks. The evaluation process should aim to determine the effectiveness of these methodologies in predicting potential scalability challenges and implementing timely performance enhancements, ensuring optimal server performance and superior user experiences.

5 Future Work

The evolving landscape of digital infrastructure continues to drive the exploration of innovative solutions and technologies aimed at enhancing server scalability and performance testing methodologies. As organizations strive to adapt to dynamic user demands and technological advancements, the following section outlines the future trends and emerging technologies that are expected to shape the landscape of server scalability and performance testing.

A. AI-Driven Testing and Automation

The integration of artificial intelligence (AI) and machine learning (ML) algorithms in scalability and performance testing is anticipated to revolutionize the testing landscape. AI-driven testing frameworks can facilitate intelligent test scenario generation, predictive analysis of system behavior under varying workloads, and automated identification of performance bottlenecks. By leveraging AI-driven testing tools, organizations can streamline the testing process, accelerate performance analysis, and proactively address potential scalability challenges, thereby ensuring optimal server performance and seamless user experiences.

B. Containerization and Microservices Architecture

The adoption of containerization technologies and microservices architecture is poised to redefine the scalability and testing paradigms, enabling organizations to develop and deploy applications in modular and scalable environments. Containerization platforms such as Docker and Kubernetes offer efficient resource

utilization, simplified application deployment, and seamless scalability, allowing organizations to create agile and adaptable server infrastructures. Embracing containerization and microservices architecture in scalability and performance testing can enable comprehensive testing of individual components, efficient resource allocation, and rapid application scaling, fostering enhanced server performance and flexible application management.

C. DevOps Integration for Continuous Testing

The integration of DevOps practices in scalability and performance testing is projected to enhance the efficiency and agility of testing processes. By integrating continuous testing practices within the DevOps pipeline, organizations can automate test execution, streamline performance monitoring, and facilitate seamless collaboration between development and operations teams. The implementation of continuous testing methodologies can enable real-time performance analysis, proactive identification of system vulnerabilities, and rapid deployment of performance enhancements, ensuring that server systems are continually optimized to meet evolving business requirements and user expectations.

D. Edge Computing and Distributed Testing Environments

The emergence of edge computing technologies and distributed testing environments is expected to revolutionize scalability and performance testing strategies, particularly for applications with stringent latency requirements. Leveraging edge computing infrastructures enables organizations to conduct distributed testing across geographically dispersed locations, facilitating the assessment of server performance under diverse network conditions and proximity constraints. By implementing distributed testing environments, organizations can simulate real-world usage scenarios, evaluate the impact of network latency on application performance, and optimize server configurations to ensure consistent and reliable service delivery across global user bases.

E. Quantum Computing for Advanced Performance Analysis

The advent of quantum computing presents unprecedented opportunities for advanced performance analysis and simulation of complex server environments. Quantum computing's exceptional computational capabilities offer the potential for simulating intricate workload scenarios, modelling intricate system behaviours, and facilitating in-depth analysis of server performance at an unprecedented scale. Integration of quantum computing technologies in performance testing can enable organizations to perform intricate performance simulations, optimize resource allocation, and anticipate future scalability requirements, fostering a new era of advanced server performance analysis and optimization.

References

- [1] S. Lehrig, H. Eikerling and S. Becker, "Scalability Elasticity and Efficiency in Cloud Computing: a Systematic Literature Review of Definitions and Metrics" in *Quality of SW Arch*, ACM, 2015.
- [2] T. Erl, Z. Mahmood and R. Puttini, *Cloud Computing: Concepts Technology & Architecture*, Prentice Hall, 2013.
- [3] G. Brataas et al., "CloudScale: Scalability Management for Cloud Systems", *ICPE Conf. on Performance Eng*, 2013.
- [4] D. Kossmann, T. Kraska and S. Loesing, "An Evaluation of Alternative Architectures for Transaction Processing in the Cloud", *Conf. on Management of data*, pp. 579-590, 2010.
- [5] W.-T. Tsai, Y. Huang and Q. Shao, "Testing the Scalability of SaaS Applications", *SOCA. IEEE*, 2011.
- [6] A. B. Bondi, *Foundations of Software and System Performance Engineering*, Addison Wesley, 2015.
- [7] R. Buyya, Y. S. Chee, and V. Srikumar, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities", *Department of Computer Science and Software Engineering, University of Melbourne, Australia*, July 2008, pp. 9.
- [8] D. Chappell, "A Short Introduction to Cloud Platforms", *David Chappell & Associates*, August 2008.
- [9] H. H. Liu, *Software Performance and Scalability: A Quantitative Approach*, Hoboken, N.J.:Wiley Publishing, 2009.

- [10] T. Atmaca, T. Begin, A. Brandwajn and H. Castel-Taleb, "Performance Evaluation of Cloud Computing Centers with General Arrivals and Service", *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 8, pp. 2341-2348, 2016.
- [11] M. Becker, S. Lehrig and S. Becker, "Systematically Deriving Quality Metrics for Cloud Computing Systems", *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, pp. 169-174, 2015.
- [12] N. R. Herbst, S. Kounev and R. Reussner, "Elasticity in Cloud Computing: What It Is and What It Is Not", *Proceedings of the 10th International Conference on Autonomic Computing (ICAC) 13*, pp. 23-27, 2013.
- [13] S. Lehrig, H. Eikerling and S. Becker, "Scalability elasticity and efficiency in cloud computing: A systematic literature review of definitions and metrics", *2015 11th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA)*, pp. 83-92, 2015.
- [14] R. Buyya, R. Ranjan and R. N. Calheiros, "InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services", *Algorithms and Architectures for Parallel Processing*, pp. 13-31, 2010.
- [15] K. Hwang, Y. Shi and X. Bai, "Scale-Out vs. Scale-Up Techniques for Cloud Performance and Productivity", *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, pp. 763-768, 2014.
- [16] H. AlJahdali, A. Albatli, P. Garraghan, P. Townend, L. Lau and J. Xu, "Multi-tenancy in Cloud Computing", *2014 IEEE 8th International Symposium on Service Oriented System Engineering*, pp. 344-351, 2014.
- [17] S. Islam, K. Lee, A. Fekete and A. Liu, "How a Consumer Can Measure Elasticity for Cloud Platforms", *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*, pp. 85-96, 2012.
- [18] U. Sharma, P. Shenoy, S. Sahu and A. Shaikh, "A Cost-Aware Elasticity Provisioning System for the Cloud", *Proceedings of the 2011 31st International Conference on Distributed Computing Systems*, pp. 559-570, 2011.
- [19] K. Hwang, X. Bai, Y. Shi, M. Li, W. G. Chen and Y. Wu, "Cloud Performance Modeling with Benchmark Evaluation of Elastic Scaling Strategies", *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 130-143, 2016.
- [20] K. Blokland, J. Mengerink and M. Pol, *Testing Cloud Services: How to Test SaaS PaaS & IaaS*, Rocky Nook, 2013.
- [21] D. Jayasinghe, S. Malkowski, J. Li, Q. Wang, Z. Wang and C. Pu, "Variations in Performance and Scalability : An Experimental Study in IaaS Clouds Using Multi-Tier Workloads", *IEEE Trans. Serv. Comput.*, vol. 7, no. 2, pp. 293-306, 2014.
- [22] D. Jayasinghe, S. Malkowski, Q. Wang, J. Li, P. Xiong and C. Pu, "Variations in Performance and Scalability when Migrating n-Tier Applications to Different Clouds", *IEEE 4th International Conference on Cloud Computing*, 2011.
- [23] J. Gao, P. Pattabhiraman, X. Bai and W. T. Tsai, "SaaS performance and scalability evaluation in clouds", *Proceedings of 2011 IEEE 6th International Symposium on Service Oriented System (SOSE)*, pp. 61-71, 2011.
- [24] A. A.-S. Ahmad and P. Andras, "Measuring the Scalability of Cloud-based Software Services", *2018 IEEE World Congress on Services (SERVICES)*, 2018.
- [25] B. Jennings and R. Stadler, "Resource Management in Clouds: Survey and Research Challenges", *J. Netw. Syst. Manag.*, vol. 23, no. 3, pp. 567-619, Jul. 2015.
- [26] J. Gao, X. Bai, W. T. Tsai and T. Uehara, "SaaS Testing on Clouds - Issues Challenges and Needs", *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering*, pp. 409-415, 2013.
- [27] E. F. Coutinho, F. R. de C. Sousa, P. A. L. Rego, D. G. Gomes and J. N. de Souza, "Elasticity in cloud computing: a survey", *Ann. des Télécommunications*, vol. 70, no. 7-8, pp. 289-309, 2015.
- [28] Y. Hu, B. Deng, F. Peng, B. Hong, Y. Zhang and D. Wang, "A survey on evaluating elasticity of cloud computing platform", *2016 World Automation Congress (WAC)*, pp. 1-4, 2016.
- [29] N. R. Herbst, S. Kounev, A. Weber and H. Groenda, "BUNGEE: An Elasticity Benchmark for Self-Adaptive IaaS Cloud Environments", *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pp. 46-56, 2015.
- [30] A. Bauer, N. Herbst and S. Kounev, "Design and Evaluation of a Proactive Application-Aware Auto-Scaler: Tutorial Paper", *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering*, pp. 425-428, 2017.
- [31] M. Beltran, "Defining an Elasticity Metric for Cloud Computing Environments", *Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools*, pp. 172-179, 2016.
- [32] J. Kuhlenkamp, M. Klems and O. Röss, "Benchmarking Scalability and Elasticity of Distributed Database Systems", *Proc. VLDB Endow.*, vol. 7, no. 12, pp. 1219-1230, Aug. 2014.

Gamification of NFT collections: Virtual NFT gallery

Jovan Ilovski[†] . Jovan Karamachoski[‡]

Email: jovan.ilovski@cse.uist.edu.mk, jovan.karamachoski@uist.edu.mk

[†] Department of Computer Science and Engineering, University of Information Science and Technology, “St. Paul the Apostle”, Ohrid, 6000, Republic of North Macedonia

[‡] Department of Communication Networks and Security, University of Information Science and Technology, “St. Paul the Apostle”, Ohrid, 6000, Republic of North Macedonia

Abstract

The rise of the popularity of Blockchain technologies, brought a cascade increase of interest for other Blockchain related products. The Non-Fungible Tokens (NFTs) represent a unique identification for digital assets present on a Blockchain system. The primary use of the NFTs are to represent digital images hosted on a dedicated web site. In most cases the environment or the visual interface for NFTs exhibition is a basic web page that is not interesting for the masses. The development of a virtual environment with increased details for the virtual reality, will leverage the overall experience of displaying and watching the NFTs collections to a level of exhibition in a gallery. In this paper we are proposing an on-chain cloudless semi-dynamic solution for virtual NFT gallery with increased security from theft, ransomware and inaccessibility of the content.

Keywords

Blockchain technology, NFT, Gamification, IPFS, Virtual Reality

1 Introduction

The main focus of the predecessor of Blockchain technologies is the media timestamping [1], which is an essential mechanism for obtaining and proving originality of a media in the legal process of Digital Right Management (DRM). With the introduction of Bitcoin [2], timestamping mechanism is used to prove the order of the transactions in the system.

A transaction in a Blockchain system is any kind of records, a static information written in the system or a digital asset transferred from one address to another. With the introduction of the Bitcoin, the first type of digital crypto asset is introduced, known as cryptocurrency. Beside the cryptocurrency, in 2012 the Mastercoin [3] Blockchain technology introduces the second most known crypto asset, known as token. The cryptocurrency is a digital alternative to the fiat currency, while the token can represent any digital substitution for a real asset. The token is a digital asset that can be used in security mechanisms, as a utility to incentivize the users or as a digital representation of a real asset.

There are two types of tokens: fungible and non-fungible tokens (NFTs). The fungible tokens are non-unique tokens representing an asset similar to a currency, used to pay for a good purchased inside the frame of the Blockchain technology where it exists. On the other hand, the non-fungible tokens are unique tokens associated with either a digital asset or real-life asset. The NFTs are of limited quantity, thus representing a unique piece of digital asset, which gives the token significant value. Thus it is feasible to connect a NFT with any type of art, media or other item from collectibles. The Quantum non-fungible token [4] released on the Namecoin Blockchain [5] in 2014, was the first digital art (image) created on a Blockchain platform. The introduction of NFT completely solves the problems of media timestamping from [1].

With the introduction of the Ethereum Blockchain technology [6] and the first implementation of smart contracts, there is a mechanism to leverage the use of tokens. Some of the most important projects related to

the NFTs are the Terra Nullius [7], Peperium [8], CryptoPunks [9], CryptoKitties [10]. The success of CryptoPunks is a motivation for the development of smart contract standard ERC-721 [11]. The standard was delivered in the early 2018, which brings steep rise of the interest for developing NFTs, especially in the gaming industry.

In 2018 the NFT marketplace started to grow and web based platforms like OpenSea [12], SuperRare [13], KnownOrigin [14] started to offer public space to trade and show-off NFTs. In late 2018, the web-based platform Maecenas [15] started to tokenize fine art paintings from Andy Warhol which was the first time that a real asset was traded on the Blockchain as NFT.

To enhance the experience of NFT collectible exhibition, the companies are going beyond the traditional web based platforms and are offering commercial platforms for creating virtual galleries for NFT exhibition. Several companies are offering proprietary solutions of Metaverse-based virtual galleries to create the three-dimensional (3D) or virtual environment for NFT galleries. Some of them are Spatial [16], Cryptovoxels [17] and Decentraland [18]. The developers are going even beyond the concept of simple galleries for exhibition of digital art and are building public on-chain places for NFT trading like Christie's 3.0 [19].

2 Virtual NFT gallery application architecture and code structure

General approach of the platforms for NFT management is to use on-chain storage of simple and lightweight data like wallet addresses and transactions, and off-chain (mainly cloud-based) storage for NFT and virtual environment storage. This approach is partially centralized and is prone to potential security threats like theft, ransomware or inaccessibility of the content. In order to overcome these potential problems, in this paper we are proposing an architecture that is completely on-chain cloudless semi-dynamic virtual environment for NFT exhibition.

The proposed architecture for the virtual NFT gallery (see Figure 1) offers a complete on-chain online solution where the user can exhibit the personal NFT collection. The simple architecture of the application uses the IPFS (InterPlanetary File System) [20] storage and Polygon Blockchain technology [21] in order to adopt the on-chain cloudless approach for building decentralized application.

In order to generate the virtual reality environment (the 3D space), the application uses the A-frame web framework [22]. A-frame defines the virtual reality environment inside an HTML file of the application, by specifying the sources for the textures, location of the elements in the scene, lighting of the scene and other components of the 3D scene. All segments of the A-frame virtual environment definitions are part of the HTML code. Beside the A-frame framework, the application uses the Web3 JavaScript libraries [23] to define the interaction between the Blockchain and the application interface.

The whole application code is set in a single index.html file containing a definition for the Cascading Style Sheets (CSS) design script of the web interface, the Web3 functions that are defining the interaction between the Blockchain and the web page, a single event listener definition for "load" event and two asynchronous function definitions for fetching and displaying the NFTs. The code also consists of A-frame scene assets initialization with definition for 10 NFT placeholder positions. The application shows the NFTs possessed by the user at the moment of application opening. If the user sells or buys new NFT, the application has to be manually refreshed in order to display the new collection. The application is completely uploaded to IPFS, and accessible through the following link

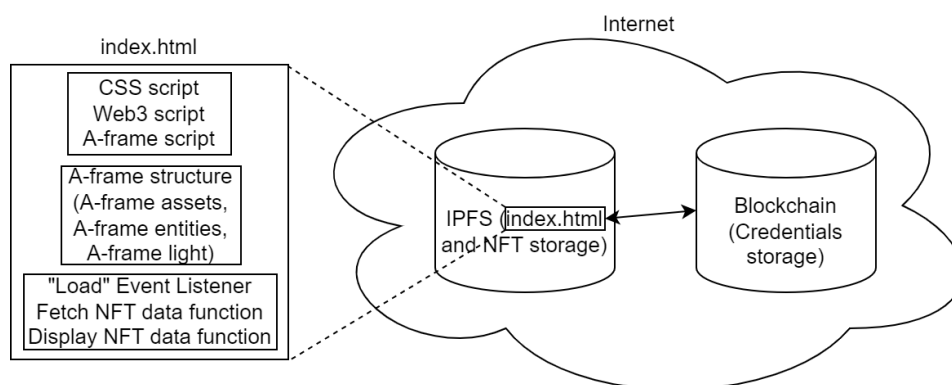


Figure 1. Architecture of the application

Figure 2 shows the simplified HTML code of the index.html file.²

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- metadata and title -->
    <style>
      /* style definition */
    </style>
    <script src="https://aframe.io/releases/1.0.4/aframe.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/web3/1.5.2/web3.min.js"></script>
  </head>
  <body>
    <a-scene>
      <a-assets>
        <!-- definition of the assets in the scene-->
      </a-assets>
      <!-- loading the references from a-assets -->
      <a-entity>
        <!-- setting up placeholder for the header text -->
      </a-entity>
      <!-- setting up 10 placeholders for the NFTs -->
      <a-entity id="nftWall" class="nft-wall" position="0 1 -4"></a-entity>
      <a-entity id="nftWall" class="nft-wall" position="0 1 -4"></a-entity>
      <a-entity id="nftWall" class="nft-wall" position="0 1 -4"></a-entity>
      <a-entity id="nftWall" class="nft-wall" position="0 1 -4"></a-entity>
      <a-entity id="nftWall" class="nft-wall" position="0 1 -4"></a-entity>
      <a-entity id="nftWall" class="nft-wall" position="0 1 -4"></a-entity>
      <a-entity id="nftWall" class="nft-wall" position="0 1 -4"></a-entity>
      <a-entity id="nftWall" class="nft-wall" position="0 1 -4"></a-entity>
      <a-entity id="nftWall" class="nft-wall" position="0 1 -4"></a-entity>
      <a-entity id="nftWall" class="nft-wall" position="0 1 -4"></a-entity>
      <!-- setting up the light of the scene -->
      <a-light type="point" intensity="2" position="2 4 4"></a-light>
    </a-scene>
  </body>
</html>
```

Figure 2. Simplified HTML code of the index.html file

The application requires Metamask [24] browser extension to interact with the Blockchain. After the initial authentication in the Metamask wallet, the user can see the NFTs that are available in the wallet. Figure 3 shows an example of the NFT gallery with NFT artwork present in a user's wallet. The user can navigate through the 3D space by use of WASD or arrow keyboard buttons. A better perspective of the artwork with additional information about it, is available by approaching closer to the desired artwork.

The current implementation of the application exists on the Mumbai testnet. It is accessible by setting up the following parameters in the Metamask extension:

- Network Name: Mumbai Testnet
- New RPC URL: <https://rpc-mumbai.maticvigil.com/>
- Chain ID: 80001
- Currency Symbol: MATIC
- Block Explorer URL: <https://polygonscan.com/>

² <https://fleek.ipfs.io/ipfs/QmU6sYcP3qPhPaEw14yzg54xEAyWLx8kwUigy6bT8RAsc8/>

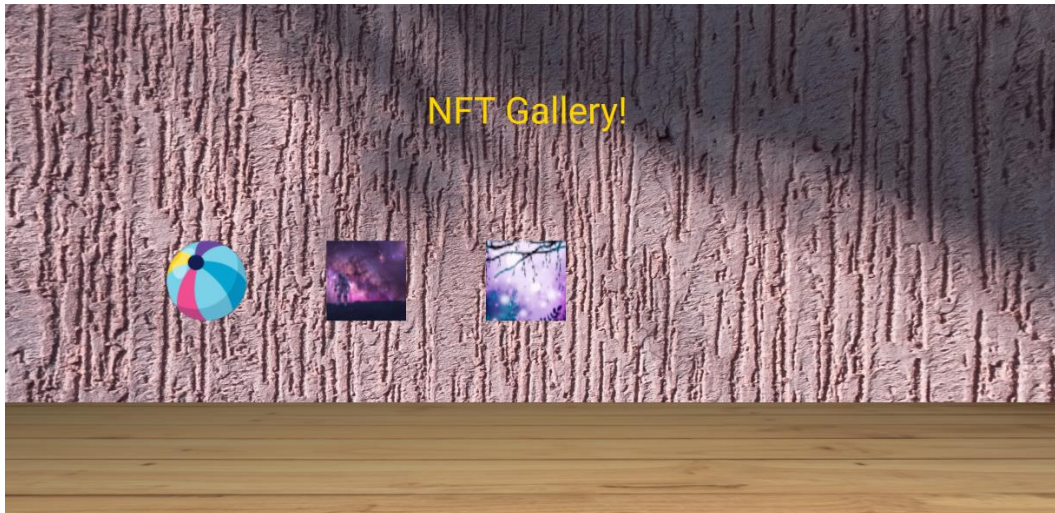


Figure 3. Example of artwork presented in the NFT gallery

The application uses the Polygon Blockchain, which is Ethereum layer 2 scaling solution that inherits the standardization procedures for writing smart contracts of the Ethereum Blockchain. The smart contract implements the ERC-721 standard for Non-Fungible Tokens, written in Solidity 0.8. The application uses the OpenZeppelin [25] templates in order to provide compatibility with the guidance and common standards for smart contract writing. Appropriate modification to the templates is conducted in order to add additional functionalities. The smart contract of the application is publicly available on the following link³.

The smart contract role is to manage the credentials of the users and allow them to mint, buy and sell NFTs. The current version of the application uses public free service nft.storage [26] (see Figure 4) to upload the artwork to the IPFS storage. Every change of NFT ownership can be updated by manually refreshing the page which gives the semi-dynamic nature of the application.

The user can interact with the smart contract and the functions defined inside the scope of the smart contract, through the Polygonscan [27] interface. Figure 5 shows an example for selling an image from the NFT collection. To proceed with selling of an artwork, the user implements the function transferFrom() where he inputs the owner's wallet address, the recipient's wallet address and the token ID for selling.

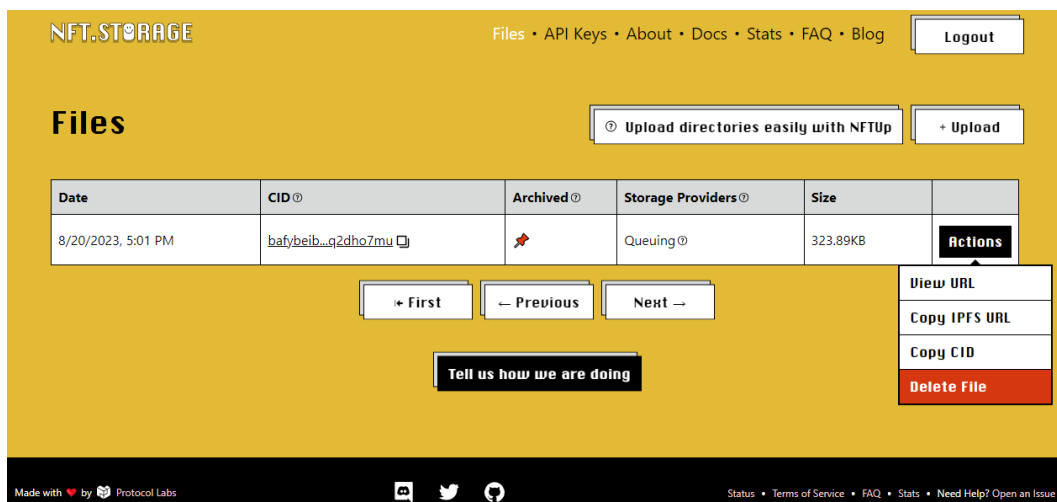
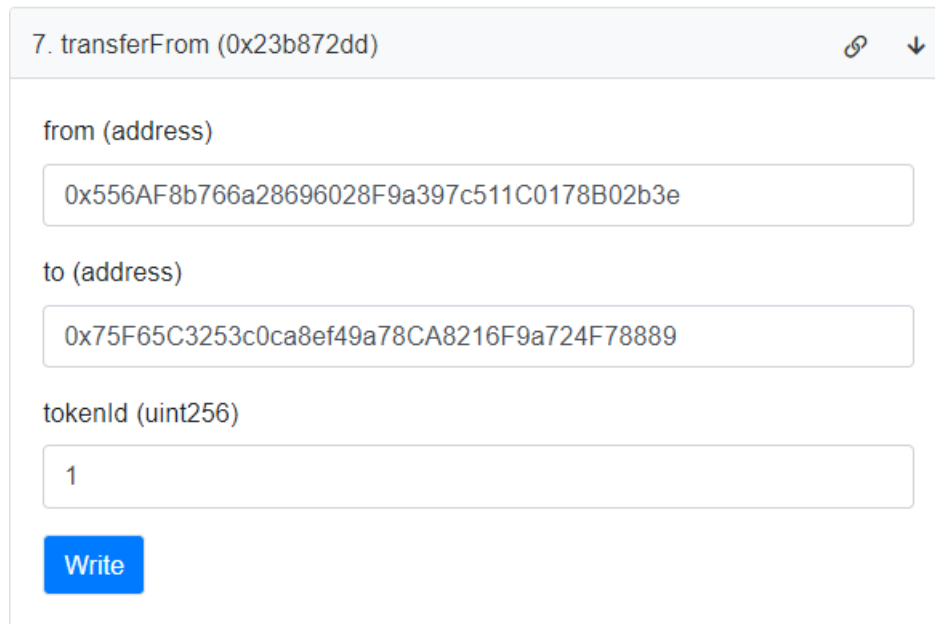


Figure 4. The uploading process of an image through nft.storage page

³ <https://mumbai.polygonscan.com/address/0x3f003e7285873e06f70ab2e5fa1c78a010de6a9e>



7. transferFrom (0x23b872dd)

from (address)

0x556AF8b766a28696028F9a397c511C0178B02b3e

to (address)

0x75F65C3253c0ca8ef49a78CA8216F9a724F78889

tokenId (uint256)

1

Write

Figure 5. Calling a smart contract function through Polygonscan

3 Advantages and disadvantages

The main advantage of using a virtual NFT gallery is the enhanced experience of the visitor, and better visibility and availability of the collectibles. Additionally, the proposed architecture has an enhanced security against theft, ransomware and inaccessibility of the artwork.

The example from Figure 3 is a simplified view of the application. The virtual environment can be enhanced in a direction to show the real environment of an actual museum or gallery. By tokenization of the art exhibited in these museums/galleries with NFTs, they can be sold as a regular digital art. This tokenization can further increase the income of the museums and galleries from selling art or selling virtual tours with virtual reality/augmented reality experience.

The main disadvantage of the proposed architecture is a result of the IPFS protocol limitations. Due to the static content addressing in the IPFS, the NFT gallery 3D space can be initiated only with a predefined number of art positions and static virtual environment definitions. Although this can be enhanced and improved by using more sophisticated frameworks and technologies, yet compromises have to be made and some kind of refreshing mechanism has to exist. Also if we want to make design changes, changes to the environment or update the code, the hash of the web-site is going to change after a push in the code, so there will be a different URL. This limitation accompanied with the requirement for manual page refresh after NFT collection modification, gives the semi-dynamic nature of the virtual environment.

4 Conclusion

The proposed architecture offers enhanced experience for the NFT collectible exhibition. The enhanced security through complete on-chain integration and cloudless mechanisms, allows the content to be theft-proof, automated, ransomware-proof and accessible at any time from everywhere. The semi-dynamic nature of the virtual environment enables the users to initialize the application with sufficient number of image positions depending on their needs. This limitation is the only disadvantage that can hinder massive adoption of this application architecture.

References

- [1] S. Haber and W. S. Stornetta, “How to time-stamp a digital document,” *Journal of Cryptology*, vol. 3, no. 2, pp. 99–111, 1991 [Online]. Available: <https://doi.org/10.1007/BF00196791>
- [2] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008 [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
- [3] J. R. Willett, “The Second Bitcoin Whitepaper,” 2012. [Online]. Available: <https://cryptochainuni.com/wp-content/uploads/Mastercoin-2nd-Bitcoin-Whitepaper.pdf>. [Accessed: 01-2024]
- [4] K. McCoy, “Quantum NFT,” 2014. [Online]. Available: <https://www.sothebys.com/en/buy/auction/2021/natively-digital-a-curated-nft-sale-2/quantum?locale=en>. [Accessed: 01-2024]
- [5] A. Loibl, “Namecoin,” 2014. [Online]. Available: https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2014-08-1/NET-2014-08-1_14.pdf. [Accessed: 01-2024]
- [6] V. Buterin and others, “Ethereum white paper: A next-generation smart contract and decentralized application platform,” *Ethereum [Online resource]*. URL: http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf, 2014.
- [7] A. McBride, “Terra Nullius — The First NFT on Ethereum,” 2021. [Online]. Available: <https://adamamcbride.medium.com/2015-the-first-nft-on-ethereum-2d2ff6468d49>. [Accessed: 01-2024]
- [8] “Peperium NFT.” [Online]. Available: <https://www.peperium.art/>. [Accessed: 01-2024]
- [9] “CryptoPunks.” [Online]. Available: <https://cryptopunks.app/>. [Accessed: 01-2024]
- [10] “CryptoKitties.” [Online]. Available: <https://www.cryptokitties.co/>. [Accessed: 01-2024]
- [11] “ERC-721 Non-Fungible Token Standard.” [Online]. Available: <https://eips.ethereum.org/EIPS/eip-721>. [Accessed: 01-2024]
- [12] “OpenSea website.” [Online]. Available: <https://opensea.io/>. [Accessed: 01-2024]
- [13] “SuperRare website.” [Online]. Available: <https://superrare.com/>. [Accessed: 01-2024]
- [14] “KnownOrigin website.” [Online]. Available: <https://knownorigin.io/>. [Accessed: 01-2024]
- [15] “Maecenas website.” [Online]. Available: <https://www.maecenas.co/>. [Accessed: 01-2024]
- [16] “Spatial website.” [Online]. Available: <https://www.spatial.io/>. [Accessed: 01-2024]
- [17] “Cryptovoxels website.” [Online]. Available: <https://www.voxels.com/>. [Accessed: 01-2024]
- [18] “Decentraland website.” [Online]. Available: <https://decentraland.org/>. [Accessed: 01-2024]
- [19] “Christie’s website.” [Online]. Available: <https://nft.christies.com/>. [Accessed: 01-2024]
- [20] J. Benet, “IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3),” *IPFS [Online resource]*. URL: <https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf>, 2014.
- [21] “Polygon Technology.” [Online]. Available: <https://polygon.technology/>. [Accessed: 01-2024]
- [22] “A-frame website.” [Online]. Available: <https://aframe.io/>. [Accessed: 01-2024]
- [23] “Web3 JS website.” [Online]. Available: <https://web3js.readthedocs.io/en/v1.10.0/>. [Accessed: 01-2024]
- [24] “Metamask website.” [Online]. Available: <https://metamask.io/>. [Accessed: 01-2024]
- [25] “OpenZeppelin website.” [Online]. Available: <https://www.openzeppelin.com/>. [Accessed: 01-2024]
- [26] “NFT Storage website.” [Online]. Available: <https://nft.storage/>. [Accessed: 01-2024]
- [27] “Polygonscan website.” [Online]. Available: <https://polygonscan.com/>. [Accessed: 01-2024]