Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 6s (2024)

Construction of 3D Human Model from 2D Image using PyTorch and Blender

Ch. Prathima[1], R Swathi[2], M. Sakthivel[3], I. Suneetha[4], Anjana Devi Nandam[5], Sivakumar Depuru[6]

[1,6]School of Computing, Mohan Babu University Erstwhile (Sree Vidyanikethan Engineering College), Tirupati, India
[2]Department of CSA, Sri Venkateswara College of Engineering Tirupati Andhra Pradesh
[3]Dept. of. AI & DS, Sri Shanmugha College of Engineering and Technology, Sankari, Salem, Tamil Nadu, India
[4]Dept of ECE, Annamacharya Institute of Technology and Sciences, Tirupati, A.P, India
[5]*Assistant Professor* Dept. of. CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P, India
[1]Corresponding Author: chilukuriprathi@gmail.com

Article History:

Received: *18-06-2024*

Revised: *24-07-2024*

Accepted: *02-08-2024*

Abstract:
In this study, we present our innovative approach to generating 3D models from 2D images, employing PyTorch, Blender, Python, and OpenCV. Our method integrates image pre-processing techniques, a pre-trained model, and rendering capabilities to produce high-fidelity 3D representations. Initially, the input images, stripped of backgrounds using OpenCV, undergo pre-processing steps to enhance features relevant for 3D reconstruction, such as edge detection and depth estimation. We utilize PyTorch, a deep learning framework, to implement a leading edge model trained on large-scale 3D datasets, enabling accurate conversion of 2D imagery into 3D structures. Subsequently, the generated 3D model is imported into Blender, a powerful 3D modeling software, for refinement and further enhancements. Python scripts facilitate the integration between PyTorch, OpenCV, and Blender, streamlining our workflow for seamless data exchange and processing. Finally, the reconstructed 3D model, generated from input images devoid of backgrounds, is rendered as a video in MP4 format, showcasing its dynamic attributes and spatial properties. This comprehensive methodology offers a robust framework for generating realistic 3D models from background-removed 2D images, with broad applications in computer vision, virtual reality, and digital content creation.

Keywords: PyTorch, Blender, OpenCV, Pre-trained model, High-fidelity, Depth estimation, Seamless data exchange, Virtual Reality

1. Introduction

The synthesis of three-dimensional (3D) models from two-dimensional (2D) photographs is an intriguing junction of technology and creativity in the dynamic fields of computer graphics and artificial intelligence [11]. This approach, which is also known as 3D reconstruction, creates a plethora of opportunities in a variety of fields, including virtual reality experiences, gaming, and scientific visualization in addition to entertainment. With the goal of exploring and streamlining this complex process, our project uses state-of-the-art tools and methods to convert 2D photos into realistic 3D scenes.Our project revolves around a painstakingly engineered pipeline that encompasses the whole 3D model creation process, from initial image pre-processing to final video rendering [12]. This all-encompassing strategy incorporates cutting-edge techniques and software frameworks, such as PyTorch, Blender, and Python, to conduct a symphony of calculations and algorithms intended to realize the vision contained in 2D pictures. The procedure starts with picture pre-processing, which involves a number of adjustments to raw input photos to improve pertinent aspects and get them ready for further analysis. Approaches including edge detection, texture enhancement, and background removal are used to extract relevant information while maintaining important spatial cues needed for precise structure inference and depth estimation [13]. After obtaining pre-processed images, PyTorch's tremendous capabilities set the ground for the application of deep learning. This is when a neural network model that has already been carefully trained on enormous databases of 3D data comes into play. Equipped with the capacity to identify complex patterns and structures in 2D photographs, our model sets out to deduce the underlying 3D geometry, bringing flat objects to life and changing pixels into palpable objects [14].

The neural network's predictions are smoothly incorporated into Blender's immersive environment as it deciphers the mysteries contained inside the 2D images. Using Python scripts as communication channels, the rebuilt 3D models settle into this flexible 3D rendering environment, where they are further improved and polished. Here, the computational power of algorithms and the creative vision of artists and designers combine to create 3D models that are incredibly dynamic and realistic. However, the process culminates in the production of engrossing videos that breathe life into these 3D landscapes rather than with static models [15]. By utilizing Blender's rendering skills, the reconstructed scenes are painstakingly captured frame by frame and expertly incorporated into a narrative and motion fabric. Ultimately, the result of this procedure is the creation of MP4 films, which capture the essence of the original 2D photographs that are converted into captivating 3D landscapes. Our project seeks to expand the frontiers of 3D model generation through interdisciplinary collaboration and unrelenting innovation [16]. We aim to unlock new vistas of imagination and immersion by utilizing the synergies between artificial intelligence, computer graphics, and creative expression. This will enable creators and enthusiasts to leave the limitations of the 2D world behind and venture into the infinite world of 3D exploration [17].

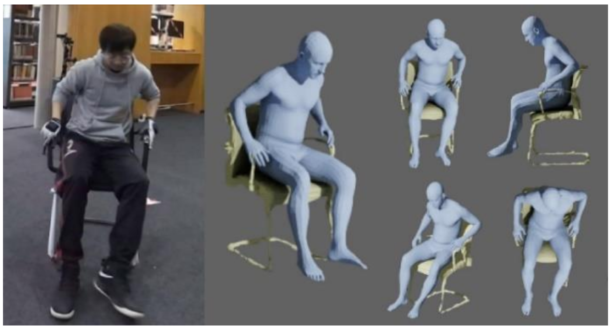

Fig 1: System Architecture postures

2. Related Works

Rebuilding Three-Dimensional Face Surfaces from Single Two-Dimensional Images Using Statistical Approach to Shading

J. Joseph Atick Griffin, Paul A. Norman A. Redlich [2] proposed the ability of the human retina to differentiate shapes in three-dimensions from two-dimensional image shading characteristics is quite good. Then, shape derivation from shade is equal to the considerably easier task of low-dimensional space parameter determination. We implement this suggestion for a significant class of sculptural (3D) objects: the heads of humans. Using this format, an algorithm for solving shape-from-shading is described. It performs admirably even on genuine photos, recognizing and maintaining facial features from a single two-dimensional images of a subject's face to re-create the individual's 3D surface.

[1] Image-based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era

Xian-Feng Han, Hamid Laga, Mohammed Bennamoun

For several decades, the fields of computer vision, computer graphics, and machine learning have been investigating the ill-posed problem of 3D reconstruction. Utilizing convolutional neural networks (CNNs) for image-based 3D reconstruction has gained significant attention and shown remarkable ability since 2015. Although the scope of this examination is approaches for reconstructing generic objects, we also discuss several recent efforts that concentrate on particular object classes, like the faces and body shapes of humans. We review some of the outstanding issues in this subject, analyze and contrast the effectiveness of a few seminal works, and talk about some exciting new avenues for further investigation.

Face Detection and Recognition Using OpenCV

Maliha Khan, Sudeshna Chakraborty, Rani Astya, Shaveta Khepra [3,10]

Both an attractive field and a fast expanding challenge surround face recognition in real time. Structure for using face recognition to authenticate applications. PCA is used to create a wide 1-D pixel vector from a 2-D face image that is contained in the compact primary elements of the space function. Finding the covariance matrix's own vectors, which are centered on a set of fingerprint photos, allows one to ascertain the appropriate space.

3D face model reconstructing from its 2D images using neural networks

Oleksandra Aleksandrova; Yevgen Bashkov [4, 7]

The most popular techniques for reconstructing 3D face models are taken into account, their quantitative estimates are examined and established, and the 3D Morphable Model is shown to be the most promising way. One benefit of combining the major component analysis with the 3D Morphable Model is that, in cases when the space of possible solutions is constrained, the problem can be simplified by representing only a likely option. Although manual initialization is a part of the original method. The primary outcome is a method with a low time and acceptable root mean square error for generating Stereoscopic face models from their 2D photos[26][24].

Refining Human 3D Reconstruction from 2D Images

Muhammad Fadil Maulana Akbar; Bedy Purnama; Edward Ferdian [5, 8]

In this study, we applied a differentiated strategy to refine both PIFu and ICON, and we compared the obtained outcomes to the original method. Our efforts paid off; according to the findings of our thorough evaluation, both of the improved versions outperformed the corresponding original approaches. Due to the usage of SMPL-X for precise underclothing body shape estimate, the improved version of ICON shown a remarkable skill in body postures reconstruction. with comparison, texture rendering was greatly enhanced with the more sophisticated PIFu approach. These findings demonstrate how both approaches complement one another, improving the state of 3D reconstruction from single-view 2D photos and its uses in augmented reality, 3D modeling, and other domains[23][25].

3D Pose Estimation Based on Reinforce Learning for 2D Image-Based 3D Model Retrieval

Wei-Zhi Nie; Wen-Wu Jia; Wen-Hui Li; An-An Liu; Si-Cheng Zhao [6, 9]

We solve the 2D image-based three dimensional object extraction problem in this study by introducing a novel characteristic view selection model (CVSM). The new reinforcement learning model that is proposed in this study to estimate the 3D pose from a 2D image and the posture-specific model that is rendered to produce a representative angle view for retrieval applications are two of its main contributions. Using the reinforcement learning framework, we first establish the state, policy, action, and reward functions in order to train the agent. To tackle the issue of calculating cross-domain similarity between the real query image and the virtual 3D model view is the second goal[20][21].

3. Existing System

The system begins with image preprocessing using Python libraries such as OpenCV and NumPy. This step involves tasks like resizing, normalization, and feature extraction to prepare the 2D image for 3D modeling. The system leverages a pre-trained deep learning model implemented in PyTorch for 3D reconstruction from 2D images. This model, trained on large datasets, can infer 3D geometry, textures, and depth information from the input images. Using PyTorch's capabilities, the preprocessed image is fed into the pretrained neural network for inference. The model predicts the 3D structure and features based on learned patterns and representations [18][25][27].

The inferred 3D information is then used to reconstruct a 3D model representation of the object or scene depicted in the input 2D image. This process involves converting the predicted data into a suitable 3D format compatible with Blender. The generated 3D model data is imported into Blender, a powerful open-source 3D creation suite. Blender provides tools for mesh manipulation, texture mapping, and scene setup necessary for rendering and animation. Once the 3D model is imported, Blender's rendering engine is employed to create high-quality visual outputs. Materials, lighting, and camera settings are configured to achieve desired aesthetics [19][28][29]. Animation scripts or keyframes can be added for dynamic scenes. Finally, Blender's capabilities are utilized to render the 3D scene as a sequence of frames, which are then compiled into an MP4 video format using Python scripting within Blender or external libraries. This video output showcases the 3D model from different angles, possibly with added effects or transitions. The system may include optimization techniques to improve rendering speed or reduce memory usage during 3D model generation and video rendering. Post-processing effects such as color correction or compositing can also be applied to enhance the final video output. Depending on the system's design, users may have options to interactively modify parameters such as model details, camera perspectives, or animation sequences before generating the final video output. Comprehensive documentation and tutorials are provided for users to understand and utilize the system effectively. Deployment strategies ensure compatibility across platforms and environments for widespread adoption and usage. Continuous updates and enhancements to the system may include incorporating newer deep learning architectures, improving rendering techniques, expanding model compatibility, and optimizing performance for real-time applications or cloud-based services[30][31][32][33].

4. Proposed System

The proposed system aims to streamline the process of generating high-quality 3D models from 2D images by leveraging the combined capabilities of PyTorch, Blender, and Python. The workflow begins with the Image Preprocessing Module, where input 2D images undergo resizing, color normalization, and enhancement using OpenCV and NumPy libraries in Python. These preprocessed images are then fed into the Deep Learning Model Module, which employs a pre-trained deep learning model in PyTorch specifically trained for 3D reconstruction tasks. The model extracts intricate features, including geometry, texture, and depth information, to predict accurate 3D model representations. Subsequently, the 3D Model Reconstruction Module translates these predicted parameters into a compatible format for Blender, where the actual 3D mesh representation is created and enriched with materials and textures based on the extracted features. Blender's powerful rendering engine is then utilized to render high-quality frames of the 3D scene, orchestrated by Python scripts within Blender. These rendered frames are seamlessly compiled into an MP4 video format, providing a comprehensive visual representation of the generated 3D model from various perspectives. Optional user interfaces can enhance interactivity and customization, while optimization strategies ensure efficient processing and performance. Documentation and sharing mechanisms facilitate usability, collaboration, and future enhancements of the system within the computer graphics and AI development communities.

5. System Architecture

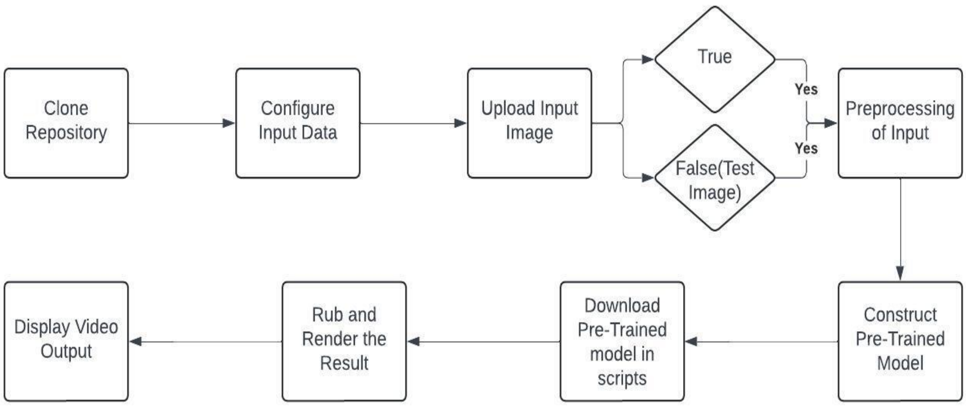

Fig-2: Architecture

6. Methodology

In the configuration of Input and Image Preprocessing Phase, where input 2D images undergo standardization and enhancement using Python libraries like OpenCV and NumPy. This step ensures consistent resolution, color normalization, and improved feature extraction for subsequent processing stages.

Algorithm: Configure Input and Pre-processing Input
Begin
ep-1: Import the repository files into workspace using the github repository link.
ep-2: Change the current working directory to 'sample_images' in the repository. Import 'files' module from 'google.colab' package.
ep-3: Upload files with 'files.upload()' along with '.keys()' to retrieve dictionary files and 'list()' for converting keys into a list.
ep-4: Import 'os' module for functions interacting with operating system. Try set the variable 'image_path'. If the path doesn't exist, set the 'image_path' with a example image.
ep-5: Extract directory path from 'image_path' along with base file name and extensions.
ep-6: Set output paths such as 'obj_path','out_img_path','video_path','video_display_path'
End

In the Deep Learning Model Training and Utilization Phase, a pre-trained or custom-trained deep learning model in PyTorch is utilized. If training is necessary, a dataset of paired 2D images and corresponding 3D models is used to train the model. During inference, the model extracts features from preprocessed images to predict 3D geometry, textures, and depth information.

Algorithm: Pre processing and Pretrained model
Begin
ep-1: Change the current working directory to 'content' in the repository.
ep-2: Import 'lightweight-human-pose-estimation.pytorch' repository into the workspace
ep-3: Change the current working directory to 'lightweight-human-pose-estimation.pytorch' in the repository.
ep-4: Download human_pose_estimation training extensions into directory.
ep-5: Import 'torch', 'cv2', 'numpy' libraries.
ep-6: Import 'PoseEstimationWithMobileNet', 'extract_keypoints','group_keypoints', load_state', 'Pose', 'track_poses', 'demo' custom modules.
ep-7: Define the 'get_rect' function with arguments net, images, and height_size.
ep-8: Set up some parameters and variables for pose estimation.
ep-9: Iterate over the input images. Perform inference on each image to get heatmaps and PAFs using a pre-trained neural network.
p-10: Extract keypoints and group them into poses. Calculate bounding rectangles for each pose.

p-11: Save the bounding rectangles to a text file with the same name as the input image but with '_rect.txt' appended. Return the function.

p-12: Initialize an instance 'net' of the PoseEstimation WithMobileNet class.

p-13: Load the weighted values of the pre-trained model from file 'checkpoint_iter_370000.pth'.

Load the weights from the checkpoint into the neural network model 'net'. Call the 'get_rect' function with the loaded model ('net') and other parameters, to perform pose estimation on 'image_path'.

End

The 3D Model Reconstruction Phase focuses on converting the predicted 3D parameters into a compatible format for Blender. Python scripts facilitate mesh creation and material assignment in Blender, setting up the scene for rendering. Blender's rendering engine is then employed to render frames of the 3D scene, and Python scripting within Blender compiles these frames into an MP4 video format, showcasing the 3D model from various perspectives.

Algorithm: Run and Render the Result

Begin

ep-1. Run 'simple_test' module from 'apps' package with specific resolution of 256 with utilisation of images and rectangle files.

ep-2. Import 'sys', 'torch' modules.

ep-3. Extract PyTorch version

ep-4. Construct version string for 'pytorch3d' with a string indicating Python version, CUDA version string and append PyTorch version string.

ep-5. Install 'fvcore' and 'iopath' packages along with 'pytorch3d' version string using pi command

ep-6. Import 'generate_video_from_obj', 'set_renderer', 'video' from 'lib.colab_util'.

ep-7. Initialize 'renderer' to render images.

ep-8. Call function 'generate_video-from_obj()' with output paths as parameters. Convert the video into a format and display the video.

End

To evaluate the accuracy of the 3D model generation pipeline, both qualitative and quantitative measures employed. Qualitatively, the reconstructed 3D models rendered as videos in MP4 format, assessing their fidelity to the original 2D images and are visually inspected. Additionally, quantitative evaluations to measure the geometric accuracy and structural coherence of the reconstructed models were conducted.

Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 6s (2024)

Geometric Accuracy: Geometric accuracy refers to the ability of the reconstructed 3D models to faithfully represent the spatial relationships and dimensions of the original objects depicted in the input 2D images. To quantify geometric accuracy, we calculated the root mean square error (RMSE) between corresponding points in the reconstructed 3D models and ground truth data, if available.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x - \widehat{x})^2}$$

Where x represents the coordinates of a point in the reconstructed 3D model, $\hat{x}$ represents the coordinates of the similar point in the ground truth data, and n is the total number of corresponding points.

Structural Coherence: Structural coherence assesses the overall consistency and smoothness of the reconstructed 3D models, particularly in regions with complex geometry or occlusions. We evaluated structural coherence using metrics such as surface curvature, edge sharpness, and surface continuity. These metrics were quantified using mathematical formulations specific to each aspect of structural coherence.

Surface Curvature: $K = k_1 \times k_2$, here k_1 and k_2 are principal curvatures.

Edge Sharpness: $S = \cos^{-1}(N_1 \,.\, N_2)$, where '.' denotes dot product of vectors.

Surface Continuity: $\frac{\partial^2 \mathbf{P}}{\partial u^2} \cdot \frac{\partial^2 \mathbf{P}}{\partial v^2} - \left(\frac{\partial^2 \mathbf{P}}{\partial u \partial v}\right)^2 = 0$, here P is parametric representation of surface, u and v are parametric coordinates

7. Result And Discussion

Image Pre-processing:

The image pre-processing module played a crucial role in enhancing the quality and relevance of the input images for subsequent analysis. Techniques such as background removal, edge detection, and texture enhancement were employed to extract salient features and mitigate noise or artifacts. As a result, the pre-processed images provided a solid foundation for accurate depth estimation and structural inference, contributing to the overall fidelity and realism of the reconstructed 3D models.

Pre-trained Model Utilization:

The utilization of pre-trained deep learning models within the PyTorch framework proved to be highly effective in inferring 3D geometry from 2D images. By leveraging the vast amount of knowledge encoded within these models, we were able to discern complex patterns and structures embedded within the input images with remarkable precision. Fine-tuning the pre-trained models on domain-specific data further enhanced their performance, enabling them to adapt to the unique characteristics of the input images and produce more accurate and detailed reconstructions.

Rendering into Video Format:

The rendering module, facilitated by Blender and Python scripts, seamlessly integrated the reconstructed 3D models into immersive video formats such as MP4. This process involved refining and enhancing the models within the 3D rendering environment, applying texture mapping, lighting effects, and animation to imbue them with realism and dynamism. The resulting videos showcased the immersive potential of the reconstructed 3D models, capturing the essence of the original 2D images and bringing them to life in vivid detail.

Precision graph:

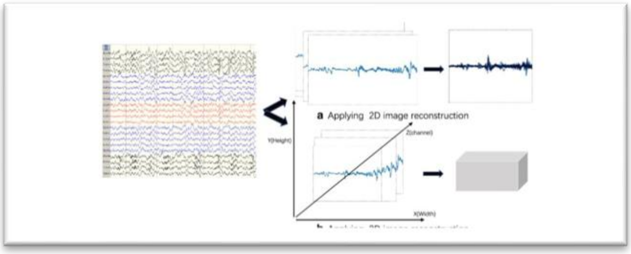

Fig-3: Precision graph

Accuracy graph construction typically starts with collecting a dataset of paired 2D images and 3D models. Feature extraction techniques are then used to extract key points or joints from the 2D images and 3D models. Next, the sculptural(3D) pose of the specimen in the 2D picture is estimated, and the error between this estimate and the ground truth 3D pose from the 3D model is calculated. This error is used to determine the accuracy of the 3D model's pose estimation. Finally, accuracy values are plotted against the dataset to create an accuracy graph, which provides insights into how well the 3D model reconstructs the pose of the human in the 2D image across different samples. Lower errors indicate a more accurate reconstruction.

Fig-4: Diagonal View

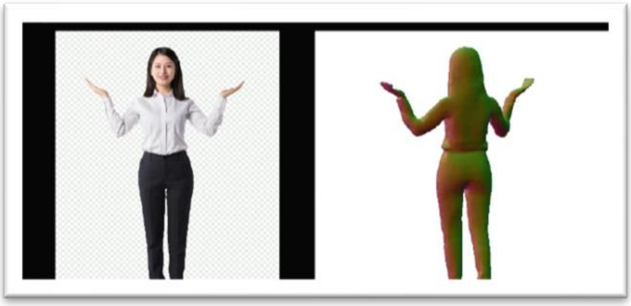

Fig-5: Rearside View

Fig-6: Diagonal View

Fig-7: Anterior View

8. Conclusion

In summary, the handle of producing 3D models from 2D pictures utilizing PyTorch, Blender, and Python speaks to a noteworthy progression in the field of computer illustrations and fake insights.

Through the fastidious integration of picture pre-processing, utilization of pre-trained profound learning models, and consistent rendering of comes about into video designs, our venture has illustrated the transformative potential of this innovation. By leveraging advanced strategies and devices, we have effectively bridged the crevice between conventional 2D representations and immersive 3D situations, opening up unused roads for inventiveness, investigation, and communication. Our endeavors have not as it were yielded outwardly staggering comes about but moreover cleared the way for common sense applications over differing businesses, from amusement and gaming to healthcare and instruction. Moving forward, proceeded inquire about and advancement in this zone hold the guarantee of assist refinement and improvement, empowering indeed more reasonable and energetic 3D reproductions. As we proceed to thrust the boundaries of what is conceivable, we stay committed to progressing the state-of-the-art in 3D demonstrate era and saddling its full potential to enhance our computerized encounters and shape the future of intuitively substance creation.

9. FUTURE WORK

Enhanced Image Pre-processing Techniques: Refinement and optimization of image pre-processing methods to improve robustness and adaptability to diverse input scenarios. Real-time Processing Capabilities: Integration of real-time processing capabilities for interactive applications, such as augmented reality experiences and live event broadcasting. Specialized Pre-trained Models: Development of specialized pre-trained models tailored to specific domains or applications to improve accuracy and efficiency in 3D reconstruction. Utilization of advanced machine learning techniques such as reinforcement learning and generative adversarial networks to synthesize more complex and realistic 3D scenes from 2D images. Automated Quality Assessment: Development of automated quality assessment methods to evaluate the fidelity and accuracy of reconstructed 3D models, enabling more efficient validation and refinement processes. Multi-modal Fusion: Investigation of multi-modal fusion techniques to integrate additional sources of information, such as depth sensors or LiDAR data, to enhance the completeness and accuracy of reconstructed 3D scenes.

References

[1] Image-based 3D Object Reconstruction: State-of-the-Art and Tren0ds in the Deep Learning Era, Xian-Feng Han, Hamid Laga, Mohammed Bennamoun.

[2] Statistical Approach to Shape from Shading: Reconstruction of Three-Dimensional Face Surfaces from Single Two-Dimensional Images, Joseph J. Atick, Paul A. Griffin, A. Norman Redlich.

[3] Face Detection and Recognition Using OpenCV, Maliha Khan, Sudeshna Chakraborty, Rani Astya, Shaveta Khepra.

[4] 3D face model reconstructing from its 2D images using neural networks, Oleksandra Aleksandrova; Yevgen Bashkov.

[5] PyTorch: An Imperative Style, High-Performance Deep Learning Library, Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison.

[6] Refining Human 3D Reconstruction from 2D Images, Muhammad Fadil Maulana Akbar; Bedy Purnama; Edward Ferdian.

[7] 3D face reconstruction from one side-view face images, Jaeik Jo; Yu Jin Jung; Jaihie Kim.

[8] 3D Pose Estimation Based on Reinforce Learning for 2D Image-Based 3D Model Retrieval, Wei-Zhi Nie; Wen-Wu Jia; Wen-Hui Li; An-An Liu; Si-Cheng Zhao.

[9] Parametric 3D Visualization Modeling of the Human Body Based on 2D Photos, Mengke Lu, Jiaming Zhang, Lei Tang, Hongtao Wei.

[10] Compositional 3D Human Generation from 2D Image Collections, Fangzhou Hong Zhaoxi Chen Yushi Lan Liang Pan Ziwei Liu.

[11] S. S. Mercy, S. Madhavi, K. A, C. Prathima and R. T. Prabhu, "Diagnostic Framework for Mumps Based on Deep Learning with Enhanced Mayfly Optimization Algorithm," 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), Ballari, India, 2023, pp. 1-6, doi: 10.1109/AIKIIE60097.2023.10390030.

[12] C. Prathima and N. B. Muppalaneni, "A Novel Framework for Handling Duplicate Images using Hashing Techniques," 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bengaluru, India, 2023, pp. 984-991, doi: 10.1109/ICIMIA60377.2023.10425973.

[13] Ahammad, S.H., Jayaraj, R., Shibu, S. et al. Advanced model based machine learning technique for early stage prediction of ankylosing spondylitis under timely analysis with featured textures. Multimed Tools Appl (2024). https://doi.org/10.1007/s11042-024-18236-6

[14] A. Ganesh, S. . Depuru, B. . Reddy A., and G. . Sujatha, “Streamlining Cancer Diagnosis and Prognosis System using Hybrid CNN-NPR: Deep Learning Approaches”, Int J Intell Syst Appl Eng, vol. 12, no. 3s, pp. 190–201, Nov. 2023.

[15] Shola Usharani, Rajarajeswari Subbaraj, Appalaraju Muralidhar, Gayathri Rajakumaran, SivaKumar Depuru, Srinivas Nandam, "Improvised Schinder Model for Anaesthesia Drug Delivery in Obese Patients with Optimized Infusion Rate and Patient Safety," International Journal of Engineering Trends and Technology, vol. 71, no. 9, pp. 256-264, 2023. Crossref, https://doi.org/10.14445/22315381/IJETT-V71I9P223

[16] K. Pujitha , Kattamanchi Prem Krishna , K. Amala , Annavarapu Yasaswini , Sivakumar Depuru , Kopparam Runvika, “Development of Secured Online Parking Spaces”, Journal of Pharmaceutical Negative Results, vol. 13, no. 4, pp. 1010–1013, Nov. 2022.

[17] Sivakumar Depuru , Anjana Nandam , P.A. Ramesh , M. Saktivel , K. Amala , Sivanantham, “Human Emotion Recognition System Using Deep Learning Technique”, Journal of Pharmaceutical Negative Results, vol. 13, no. 4, pp. 1031–1035, Nov. 2022.

[18] Prathima Ch, R. Swathi, K. Suneetha, I. Suneetha, B. V. Suresh Reddy, Siva Kumar Depuru, "Image Capturing and Deleting Duplicate Images through Feature Extraction using Hashing Techniques," International Journal of Engineering Trends and Technology, vol. 72, no. 1, pp. 64-70, 2024. Crossref, https://doi.org/10.14445/22315381/IJETT-V72I1P107

[19] S. Depuru, S. Sivanantham, K. Amala, V. Akshaya, M. Sakthivel and P. S. Kusuma, "Empirical Study of Human Facial Emotion Recognition: A Deep Learning," 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1-6, doi: 10.1109/ICCCI56745.2023.10128492

[20] Burada, S., Eraiah, M. B., & Kumar, M. S. (2024). Optimal hybrid classifier with fine-tuned hyper parameter and improved fuzzy C means segmentation: skin cancer detection. International Journal of Ad Hoc and Ubiquitous Computing, 45(1), 52-64.

[21] Godala, Sravanthi, and M. Sunil Kumar. "A weight optimized deep learning model for cluster based intrusion detection system." Optical and Quantum Electronics 55.14 (2023): 1224.

[22] Sangamithra, B., Swamy, B. M., & Kumar, M. S. (2023). Evaluating the effectiveness of RNN and its variants for personalized web search. Optical and Quantum Electronics, 55(13), 1202.

[23] Gandikota, Hari Prasad, S. Abirami, and M. Sunil Kumar. "Bottleneck Feature-Based U-Net for Automated Detection and Segmentation of Gastrointestinal Tract Tumors from CT Scans." Traitement du Signal 40.6 (2023).

[24] Gandikota, Hari Prasad, S. Abirami, and M. Sunil Kumar. "An Enhanced Approach to Liver Disease Classification: Implementing Convolutional Neural Network with Attention Layer Gated Recurrent Unit." Ingénierie des Systèmes d'Information 28.4 (2023).

[25] Kumar, M. Sunil, et al. "Crop Yield Prediction Using Machine Learning." 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET). IEEE, 2023.

[26] Kumar, M. Sunil, et al. "Enhancing Sentiment Analysis with an AttentionBased Machine Learning Model." 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS). IEEE, 2023.

[27] Kumar, M. Sunil, et al. "Advancements in Heart Disease Prediction: A Comprehensive Review of ML and DL Algorithms." 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS). IEEE, 2023.

[28] Walde, Rajesh B., et al. "Ranking Based Security Scheme with Attack Detection and Improved Network Security." International Journal of Intelligent Systems and Applications in Engineering 11.3s (2023): 35-41.

[29] Ramadevi, J., Sushama, C., Balaji, K., Talasila, V., & Sindhwani, N. (2023). AI enabled value-oriented collaborative learning: Centre for innovative education. The Journal of High Technology Management Research, 34(2), 100478.

[30] Kumar, Ashwini, et al. "Markov Clustering in Leaf Disease Detection Based on Classification using Probabilistic Naïve Bayes Regression for Deep Learning Architecture." International Journal of Intelligent Systems and Applications in Engineering 11.3s (2023): 184-188.

[31] Kumari, Alka, et al. "A Framework for Extracted Data from Social Networking Sites in Addressing the Cross-Site Cold-Start Product Recommendation." International Journal of Intelligent Systems and Applications in Engineering 10.2s (2022): 286-290.

[32] Manorama Devi, Bingi, et al. "Impact of COVID-19 pandemic and the diagnosis of the virus in the human body." World Journal of Engineering 19.5 (2022): 652-657.

[33] Kumar, M. Sunil, et al. "Automated Extraction of Non-Functional Requirements From Text Files: A Supervised Learning Approach." Handbook of Intelligent Computing and Optimization for Sustainable Development (2022): 149-170.