Increased frame rate for Crowd Counting in Enclosed Spaces using GANs

Adriano Puglisi[1], Francesca Fiani[1] and Giorgio De Magistris[1]

[1]Sapienza University of Rome, via Ariosto 25, Rome, Italy

Abstract

An efficient computer system for regulating and monitoring the density of people in confined areas is very helpful. It becomes imperative to implement a solution that takes into account the processing power and pre-installed hardware in these places. Using computer vision, in particular, to make use of regular CCTV cameras that have been augmented by neural networks, solves the problem of precisely counting individuals in enclosed spaces. We describe a control system specifically designed for this goal, maximizing the capabilities of current infrastructure and enhancing neural networks to achieve higher frame rates.

Keywords

Computer vision, Tracking, YOLO, SORT, Generative Adversarial Network

1. Introduction

In many enclosed spaces, crowd capacity management is a common challenge due to strict occupancy limits. These limits are critical for safety and regulatory compliance. To address this issue, we propose to leverage CCTV cameras as a solution to more accurately count people within a confined area. Leveraging advanced video analytics, our system aims to provide real-time monitoring, helping companies and institutions maintain optimal audience density and ensure a safe environment [1, 2]. The main solutions proposed in recent years for indoor human tracking use cameras with depths for the acquisition of the position, however, this technology is in some cases expensive or in any case not available. The use of modern algorithms in computer vision allows the development of systems capable of using a simple two-dimensional camera also to calculate the depth and therefore the position of some objects, or people, in space [3]. These types of cameras usually have poor FPS values to save storage space, this tool is combined with a neural network based on the GAN framework, to increase the frame rate of such cameras. The interpolation of frames through the use of neural networks is an important and complex problem to solve, the datasets used are often very large and the networks very deep. These networks, even if they achieve remarkable results, have a very high computational cost and can often be trained only on expensive or unavailable hardware. For this reason, we choose to bias the neural network using a specific dataset for the task, that contains only working pedestrians, to obtain a faster convergence of our network. In the last few years, the GAN framework [4, 5, 6] brought a little revolution in the neural networks field. It's possible to adapt such a framework to a series of different tasks, in particular, it's broadly used in the Super-Resolution of signals, such as images, videos, and audio and, generally speaking, in recreating or reconstructing parts of lost signals. Given the potential of this framework, we decided to implement a GAN regarding the frame-rate increase of CCTV. The whole project tries to exploit the best techniques that require a saving of hardware resources, thus allowing it to be used in as many environments as possible and with a medium-low computing power. The security in closed spaces and the tracking of people are having an ever greater impact on the management of common spaces and crowded places, the use of advanced IT systems can allow greater, more effective, and efficient control. Maintaining a significant trade-off between the necessary hardware resources and the results obtained was an important point in developing our work.

ICYRIME 2023: 8th International Conference of Yearly Reports on Informatics, Mathematics, and Engineering. Naples, July 28-31, 2023

puglisi@diag.uniroma1.it (A. Puglisi); fiani@diag.uniroma1.it (F. Fiani); demagistris@diag.uniroma1.it (G. D. Magistris)

0009-0007-6307-7194 (A. Puglisi)

2. Related Works

2.1. Human tracking

The problem of human tracking and positioning is a well-known subject in computer vision. It can be useful in different situations, such as crowd control, monitoring public areas, security, and so on [7, 8, 9, 10, 11]. We want to focus on the usage in an indoor environment mainly. Some research [12] uses top-view depth cameras, subtracting the average obtained image, consisting of the floor and the furniture, segmenting the moving objects, and trying to match them with a top-view model of a person. After that the projection distortion is corrected, obtaining the position on the plane. Similarly in [13] fish-eye top-view cameras segment the moving object from the static background using adaptive GMM and correcting the projective distortion to find the position. Even

though those approaches could be effective, we want to use cameras that are usually positioned on the wall instead of the ceiling. Other papers [14] use 3D cameras to obtain an ortho-image to find objects in a scene; while this approach could be extended to our needs, it requires more sophisticated cameras with depth vision, which CCTV cameras are not equipped with.

2.2. Frame-rate increase

The computer vision community has given significant attention to the necessity of increasing the frame rate and, consequently, the video frame interpolation. Many uses for this issue exist, including the creation of slow motion and frame recovery for video streaming and gaming. High-frame rate videos are visually more pleasing to watch because they may avoid typical glitches like temporal jittering and motion blurriness. Several techniques have been used to overcome the issue of getting intermediate frames from a limited collection, including frame interpolation and, more recently, DNNs. In Frame Interpolation techniques, intermediate frames are generated between the present frames using interpolation, as in the methods proposed by Choi et al. [15], based on Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation. Also, a wide variety of DNN methods were proposed; recently Flow-Agnostic Video Representations for Fast Frame Interpolation [FLAVR [16]] solved the problem using an autoencoder based on 3D space-time convolutions, to enable end-to-end learning and inference. With no extra inputs needed in the form of depth maps or optical flow, this technique effectively learns to reason about non-linear movements, complicated occlusions, and temporal abstractions, leading to enhanced performance. Depth-Aware Video Frame Interpolation [17] is another notable DNN technique that synthesizes intermediate flows that sample items closer to the viewer preferentially by introducing a depth-aware flow projection layer. To synthesize the output frame, this approach uses the optical flow and local interpolation kernels to warp input frames, depth maps, and contextual features. Hierarchical features are utilized to extract contextual information from nearby pixels.

3. Proposed method

In this section, we describe the methods and the algorithms used to analyze the images and detect people inside the scene, and after that increase the frame rate.

3.1. Detection

YOLO [18] is the neural network framework we used for detecting persons in the scene, it is extremely popular and widely used in computer vision for its speed and accuracy in the detection. We tested a set of YOLO pre-trained models, to pick up the most suitable one for our goal. Our goal was to achieve good accuracy while maintaining a reasonable number of FPS to work with in real time. The models we tested are trained on a custom public dataset specific for crowded human places [19]. The models we tested are:

- YOLOv8n trained with 416×416 images
- YOLOv8s trained with 416×416 images
- YOLOv8m trained with 416×416 images

3.2. Tracking

To track people in the scene as reliably as possible, it's needed a good balance between accuracy and speed; while the chosen model offers a good speed in the detection, it lacks accuracy. To make up for this lack, we corrected and smoothed the predictions made by YOLO using the SORT algorithm [20], which corrects and smooths the position of the bounding boxes using a Kalman filter [21].
The Hungarian algorithm is utilized to monitor every detection inside a scene. A list of detections is stored, the positions of the detections are predicted using the Kalman filter for each iteration, the Intersection over Union (IOU) is calculated using an updated set of detections, the Hungarian algorithm is used to find the best matches, and the detections are categorized as matched or unmatched. For every bounding box, a new Kalman filter is created in case of mismatched detections. The algorithm updates the Kalman filter for matching detections. Ultimately, a list of tagged detections is produced. The state used for the Kalman filter is defined as:

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]$$

where u and v represent the horizontal and vertical positions in pixels, and s and r denote the scale (area) and aspect ratio of the bounding box. Notably, the aspect ratio lacks a corresponding velocity in the state, as it is assumed to be constant.

3.3. Spatial Localization

This section outlines the approach for obtaining the camera matrix and the algorithm employed to determine the 2D position of a person in the scene.

3.3.1. Camera Model

The finite projective camera, denoted as P, is characterized by its intrinsic and extrinsic parameters, given by:

$$P = [M| - M\tilde{C}] = K[R| - R\tilde{C}]$$

Here, R describes the orientation of the camera and $\tilde{C}$ is the world position of the camera center. K is the calibration matrix and since the resolution is the same in both the x and y directions, the calibration matrix can be defined as:

$$K = \begin{bmatrix} f & s & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

with f being the focal length andcan be obtained using the formula:

$$f = \frac{resolution}{2 * atan2(\frac{AFOV}{2})}$$

Where $AFOV$ is the field of view. Typically, obtaining these parameters requires camera calibration using methods like Zhang's method [22]. However, in a simulator environment, all parameters can be derived from the properties of the involved objects.

3.3.2. Inverting Projective Transformation

Summarizing, the 3×4 camera matrix P transforms image coordinates $(u, v, 1)^T$ to scene coordinates $(X, Y, Z, 1)^T$. To obtain the scene coordinates from image coordinates, we aim to invert P, considering that perspective projection is not injective. Assuming knowledge of the distance from the ground (height of the person), we utilize the pseudo-inverse P^+ of P. Two points on the back-projected ray are identified: the camera center C and the point P^+x. The ray is expressed as:

$$X(\lambda) = P^+x + \lambda C$$

For a finite camera with $P = [M|p_4]$, the camera center is $\tilde{C} = -M^{-1}p_4$. Back-projection of an image point x intersects the plane at infinity at the point $D = ((M^{-1}x)^T, 0)^T$, providing a second point on the ray. The line is represented as:

$$X(\mu) = \begin{pmatrix} M^{-1}(\mu x - p_4) \\ 1 \end{pmatrix}$$

Solving for μ, considering the Z coordinate as the detected height, allows computation of the X and Y coordinates in the scene.

3.4. Enhancing Frame Rate

We decided to implement a GAN solution for our framework, based on the Image2Image work [23]. The framework is composed of two models: a generator and a discriminator; the generator takes as input the frames x_t and x_{t+1} and tries to infer the missing frame y_{gen}, while the discriminator takes the same input concatenated either with the real missing frame y_{gold} or with the generated one, to classify them as generated or real. The goal is to train them at the same time, improving their performances to obtain a good model that generates the missing frames.

3.4.1. Network architecture

The generator takes as input two pictures of size $(RES \times RES \times 3)$. To minimize its dimensions, the encoder employs two-dimensional convolutional layers with a stride of two using a UNet [24]. LeakyReLU is the activation function, and its slope is 0.2. On the other hand, the decoder uses the LeakyReLU activation function with a slope of 0.2 and consists of several 2-dimensional convolutional layers with a stride of 2. The $tanh$ activation function is used in the final output layer to make sure that the outputs are inside the $[-1, 1]$ range. The same input as the generator, concatenated with the produced output y_{gen} or the genuine frame y_{gold}, is fed into the discriminator, which is built like a CNN. Table 1 summarizes the architecture.

3.4.2. Loss function

Within our generative adversarial network (GAN), an adversarial discriminator D seeks to maximize the objective function, while the generator G strives to decrease it, resulting in a zero-sum game. The definition of the objective function is:

$$\mathcal{L}_{cGAN}(G, D) =$$

$$= \mathbb{E}_{x,y}[logD(x, y)] + \mathbb{E}_{x,z}[log(1 - D(x, G(x, z)))]$$

The optimal generator denoted as G^* is determined by:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D)$$

We enhance the GAN objective function by adding the L1 loss function, which is a conventional loss. The generator's job is now to provide nearly optimum outputs using this conventional loss function, in addition to tricking the discriminator, without changing the discriminator's duty. The L1 loss, denoted as $\mathcal{L}_1$ is defined as:

$$\mathcal{L}_1(G) = \mathbb{E}_{x,y,z}[||y - G(x, z)||]$$

And now our final objective function is:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_1(G)$$

Here λ serves as a weighting parameter for the $\mathcal{L}_1$ loss.

4. Implementation

In this section we will describe the implementation details of our work, starting with the setup and the preparation of the simulator, the training phase of the neural network, and the whole system architecture.

Table 1

GAN Network architecture

Layer	Activation	Filters	Filter Size	Stride	Batch Norm
Generator					
Input	-	-	-	-	-
Conv	LeakyReLU	128	(4,4)	(2,1)	No
Conv	LeakyReLU	64	(4,4)	(2,1)	Yes
...	...	...	...	...	...
Conv	Tanh	3	(4,4)	(2,1)	Yes
Discriminator					
Input	-	-	-	-	-
Conv	LeakyReLU	16	(4,4)	(2,1)	No
Conv	LeakyReLU	32	(4,4)	(2,1)	Yes
...	...	...	...	...	...
FC	-	1	-	-	-

4.1. Language and Libraries

The whole project was developed using Python v3.8.10. For the detection and tracking part the following libraries were used:

- OpenCV v4.5.2 compiled from source, to activate the ability to use CUDA drivers and CUDNN, obtaining faster results with YOLO.
- Numpy v1.21.4

For the neural network creation, training, and testing we used:

- TensorFlow v2
- Keras for the creation of the layers
- OpenCV for the pre-processing of the dataset and the data augmentation
- Matplotlib to visualize our results

4.2. Net training and testing

For training our network, we used the EPFL [25] dataset, which includes multiple scenes of moving pedestrians. The training data were extracted by taking 3 frames at a time and adding noise to increase the number available. Next, each triplet was saved in a file, with the first and last frames as input to the generator and the middle frame as reference. The dataset was divided into validation, training, and testing. The GAN network was trained using the early stopping technique thus preventing the network from overfitting the data. The loss graph is shown in Figure 1 for the Generator and Figure 2 for the Discriminator.

To study the results of our neural network, we computed the SSIM and PSNR values which are used to measure the similarity between two images and are defined as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Where μ_x the average of x; μ_y the average of y; σ_x^2 the variance of x; σ_y^2 the variance of y; σ_{xy} the covariance of x and y; $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ two variables to stabilize the division with weak denominator; L the dynamic range of the pixel-values (typically this is $2^{\#bits\ per\ pixel} - 1$); $k_1 = 0.01$, $k_2 = 0.03$ by default.

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX\ \{I\}}{\sqrt{MSE}}\right)$$

Where $MAX\ \{I\}$ is the maximum possible pixel value of the image and with the mean square error (MSE) defined as:

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\|I(i,j) - K(i,j)\|^2$$

Let I represent the original image and K denote the generated image, both of dimensions MxN. The results of our network, in comparison with other methodologies, are presented in Table 2.

Table 2

Results compared with S.O.T.A networks

Net	SSIM	PSNR
EpicFlow[26]	0.93	31.6
BeyondMSE[27]	0.92	32
MCnet+RES[28]	0.91	31
Our Network	**0.92**	**33.2**

4.3. System architecture

This system can also be used with multiple cameras; when working with multiple cameras, each camera receives an image and elaborates that using YOLO and SORT, to extract the bounding boxes positions. Each

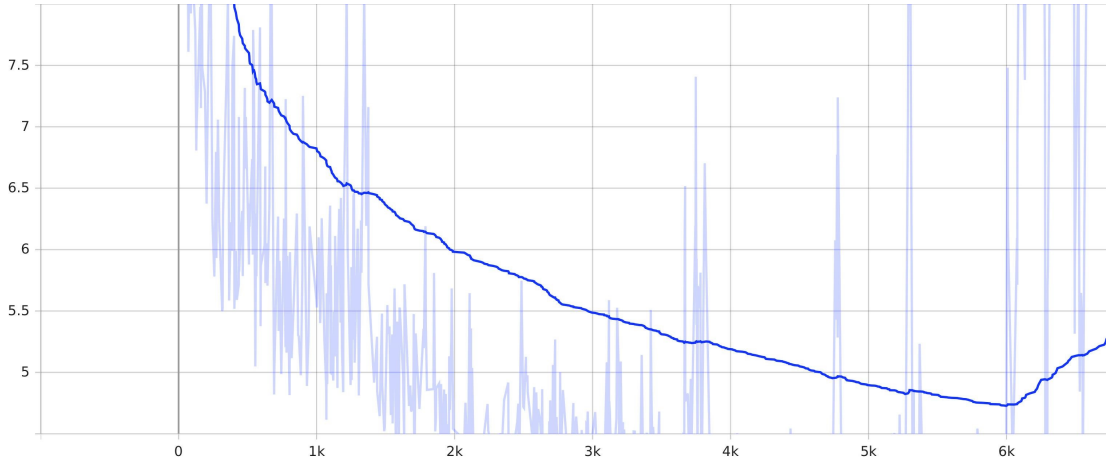

Figure 1: Generator Loss

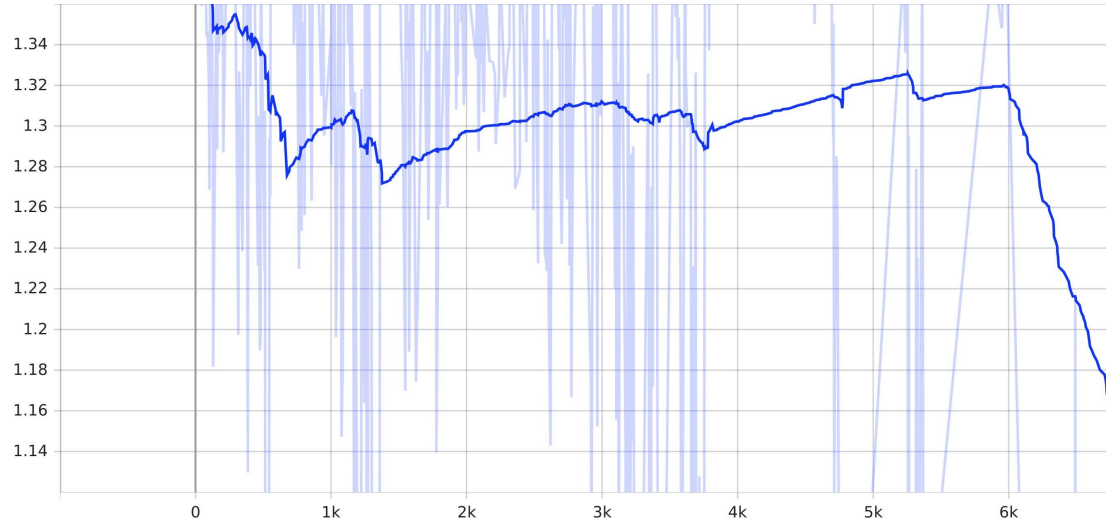

Figure 2: Discriminator Loss

frame is passed to the detection thread and can be stored, to be processed later by the Neural Network. The points centered in the top part of the bounding boxes generated by the detection threads are passed to the camera models, to obtain the position of the persons on the plane. Those positions are then merged by searching for each camera the nearest neighbor and in case of a mismatch between the number of people in the cluster, the bigger one is chosen; after matching is found, for each person, a dot is drawn on the map having the average position between the matched one. The whole system architecture is represented in the Figure 3.

5. Results

In this section, we will show the results obtained.

5.1. Frame Rate and Crowd Counting

As we can see in figure 4, the first and the last frame are the input, while the middle one was generated by the Generator of the GAN network.

After a series of comprehensive tests, our technology performed smoothly when properly identifying and counting people in enclosed spaces. With the addition of computer vision algorithms and the advances made possible by our improved neural network, accurate peo-

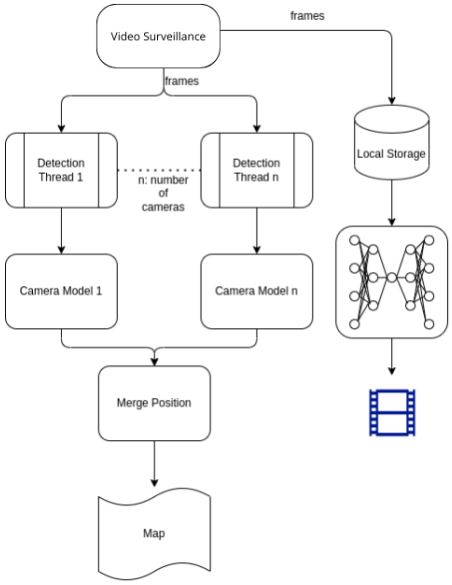

Figure 3: System architecture when working with multiple cameras

Figure 4: Example of Frame Rate Increase using our GAN

ple counting and identification are guaranteed. The outcomes demonstrate the system's capacity to monitor and control crowd density in confined areas efficiently. For a visual depiction, Figure 5 shows how our model could be used in a real-case scenario using only one camera.

6. Conclusions

In summary, our methodology offers a dependable and precise means of detecting and measuring human beings in enclosed spaces. By utilizing the creative fusion of GAN-based networks and the effectiveness of lightweight YOLO models, our system not only ensures robustness but also demonstrates flexibility to operate on systems with limited technological resources. This clever approach strengthens security protocols and expedites operational workflows in addition to offering a financially sensible way to implement occupancy restrictions in a variety of scenarios. Our method, which makes use of cutting-edge AI technologies, is a big step toward improving space management and guaranteeing adherence to safety laws, making all people's surroundings safer and more effective. Moreover, it is a useful tool in circumstances where precisely counting people is necessary to avoid crowding, making the environment safer and more effective for everyone. Our method, which makes use of cutting-edge AI technologies, is a big step toward improving space management and guaranteeing adherence to safety rules, which will eventually improve the general standard of public areas and facilities.

Figure 5: Visual representation of the system's performance in counting people within an enclosed space. The number of people detected is on the left top corner.

References

[1] N. N. Dat, V. Ponzi, S. Russo, F. Vincelli, Supporting impaired people with a following robotic assistant by means of end-to-end visual target navigation and reinforcement learning approaches, volume 3118, 2021, pp. 51 – 63.

[2] V. Ponzi, S. Russo, V. Bianco, C. Napoli, A. Wajda, Psychoeducative social robots for an healthier lifestyle using artificial intelligence: a case-study, volume 3118, 2021, pp. 26 – 33.

[3] G. De Magistris, R. Caprari, G. Castro, S. Russo, L. Iocchi, D. Nardi, C. Napoli, Vision-based holistic scene understanding for context-aware human-robot interaction 13196 LNAI (2022) 310 – 325. doi:10.1007/978-3-031-08421-8_21.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[5] S. Pepe, S. Tedeschi, N. Brandizzi, S. Russo, L. Iocchi, C. Napoli, Human attention assessment using a machine learning approach with gan-based data augmentation technique trained using a cus-

tom dataset, OBM Neurobiology 6 (2022). doi:10.21926/obm.neurobiol.2204139.

[6] C. Ciancarelli, G. De Magistris, S. Cognetta, D. Appetito, C. Napoli, D. Nardi, A gan approach for anomaly detection in spacecraft telemetries 531 LNNS (2023) 393 – 402. doi:10.1007/978-3-031-18050-7_38.

[7] N. Brandizzi, S. Russo, G. Galati, C. Napoli, Addressing vehicle sharing through behavioral analysis: A solution to user clustering using recency-frequency-monetary and vehicle relocation based on neighborhood splits, Information (Switzerland) 13 (2022). doi:10.3390/info13110511.

[8] V. Marcotrigiano, G. D. Stingi, S. Fregnan, P. Magarelli, P. Pasquale, S. Russo, G. B. Orsi, M. T. Montagna, C. Napoli, C. Napoli, An integrated control plan in primary schools: Results of a field investigation on nutritional and hygienic features in the apulia region (southern italy), Nutrients 13 (2021). doi:10.3390/nu13093006.

[9] A. Alfarano, G. De Magistris, L. Mongelli, S. Russo, J. Starczewski, C. Napoli, A novel convmixer transformer based architecture for violent behavior detection 14126 LNAI (2023) 3 – 16. doi:10.1007/978-3-031-42508-0_1.

[10] M. Woźniak, D. Połap, M. Gabryel, R. K. Nowicki, C. Napoli, E. Tramontana, Can we process 2d images using artificial bee colony?, volume 9119, 2015, pp. 660 – 671. doi:10.1007/978-3-319-19324-3_59.

[11] S. Russo, C. Napoli, A comprehensive solution for psychological treatment and therapeutic path planning based on knowledge base and expertise sharing, volume 2472, 2019, pp. 41 – 47.

[12] T.-E. Tseng, A.-S. Liu, P.-H. Hsiao, C.-M. Huang, L.-C. Fu, Real-time people detection and tracking for indoor surveillance using multiple top-view depth cameras, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 4077–4082. doi:10.1109/IROS.2014.6943136.

[13] R. Hartmann, F. Al Machot, P. Mahr, C. Bobda, Camera-based system for tracking and position estimation of humans, 2010, pp. 62–67. doi:10.1109/DASIP.2010.5706247.

[14] M.-A. Mittet, T. Landes, P. Grussenmeyer, Localization using rgb-d cameras orthoimages, volume XL-5, 2014. doi:10.5194/isprsarchives-XL-5-425-2014.

[15] B.-D. Choi, J.-W. Han, C.-S. Kim, S.-J. Ko, Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation, IEEE Transactions on Circuits and Systems for Video Technology 17 (2007) 407–416. doi:10.1109/TCSVT.2007.893835.

[16] T. Kalluri, D. Pathak, M. Chandraker, D. Tran, FLAVR: flow-agnostic video representations for fast frame interpolation, CoRR abs/2012.08512 (2020). URL: https://arxiv.org/abs/2012.08512. arXiv:2012.08512.

[17] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, M.-H. Yang, Depth-aware video frame interpolation, 2019. URL: http://arxiv.org/abs/1904.00830.

[18] M. Sohan, T. Sai Ram, R. Reddy, C. Venkata, A review on yolov8 and its advancements, in: International Conference on Data Intelligence and Cognitive Informatics, Springer, 2024, pp. 529–545.

[19] K. D. Team, Crowdhuman dataset, https://universe.roboflow.com/keio-dba-team/crowdhuman-nur7g, 2022.

[20] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, 2016 IEEE International Conference on Image Processing (ICIP) (2016). URL: http://dx.doi.org/10.1109/ICIP.2016.7533003. doi:10.1109/icip.2016.7533003.

[21] R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, Journal of Basic Engineering 82 (1960) 35–45. URL: https://doi.org/10.1115/1.3662552. doi:10.1115/1.3662552.

[22] Z. Zhang, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 1330–1334. doi:10.1109/34.888718.

[23] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[24] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, CoRR abs/1505.04597 (2015). URL: http://arxiv.org/abs/1505.04597. arXiv:1505.04597.

[25] F. Fleuret, J. Berclaz, R. Lengagne, P. Fua, Multicamera people tracking with a probabilistic occupancy map, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 267–282. doi:10.1109/TPAMI.2007.1174.

[26] J. Revaud, P. Weinzaepfel, Z. Harchaoui, C. Schmid, Epicflow: Edge-preserving interpolation of correspondences for optical flow, 2015. arXiv:1501.02565.

[27] M. Mathieu, C. Couprie, Y. LeCun, Deep multi-scale video prediction beyond mean square error, 2016. arXiv:1511.05440.

[28] R. Villegas, J. Yang, S. Hong, X. Lin, H. Lee, Decomposing motion and content for natural video sequence prediction, 2018. arXiv:1706.08033.