FPS Killer: A Moving Target Detection Algorithm in FPS Game Based on Yolov3 Frame and Gradient Descent Algorithm

\*Note: Sub-titles are not captured in Xplore and should not be used

Huang Jiayu\*  
School of Computing  
Jiangsu Vocational College of CommerceJiangsu, China 18552636796@163.com Liu Aofan   
School of Computing& Data Science  
Xiamen University MalaysiaSepang, Malaysia  
SWE2009510@xmu.edu.my

*Abstract*—With the continuous development of the economic level of the society, the game industry is getting more and more extensive development among young people. As one of the mainstreams in the game industry, FPS games are warmly welcomed by the majority of players. In order to ensure the fairness of the game, anti-cheat technologies have emerged, such as “BattlEye” and “ValveAnti-Cheat”. This article applies the current popular computer vision technology to FPS games and proposes a cheating technology that cannot be detected by traditional anti-cheating methods. However, this project will only be used for academic research and discussion, and it is forbidden to use it for commercial sales and esports.

Keywords—FPS; YOLOv3; Object Detection; Alpha-Beta Pruning

# Introduction (*Heading 1*)

FPS game stands for First Person Shooting Game, and its goal is to simulate as realistic a design experience as possible and make people and make people have an immersive experience on the battle field. Usually, it appears in this way:

In an FPS, no matter what the game is, the main aim is to kill the enemy. This enemy may be another player, or may be computer-controlled bots. But no matter what it is, it takes on a human form.

At the same time, computer vision technology represented by CNN, DNN and RNN has been widely used in machine learning. Whether in face recognition or in the field of motion capture, they have made great process in recent years [1].

The most important thing in FPS game is face recognition and motion capture. If you're standing face to face with the enemy, who can react faster and aim better has an advantage. However, as we played the game, we found it difficult to aim at enemies, especially when they were hiding somewhere or popping out. No matter in what kind of FPS game, you can always find the characters in it. Here is an example of attributes character structure in FPS game.

1. Attributes of Characters in FPS Game

|  |  |
| --- | --- |
|  | **Attributes** |
| **Position** | Position (X, Y) |
| **Velocity** | Velocity(X): Speed in X axis  Velocity(Y): Speed in Y axis  Velocity(Z): Speed in Z axis |
| **Acceleration** | Acceleration(X): a in X axis  Acceleration(Y): a in Y axis  Acceleration(Z): a in Z axis |

Hence, we came up the idea of automatic aiming. In all FPS games, we can locate the enemy's coordinates on the screen through face recognition, and then move the mouse to the corresponding position automatically by calling the Windows Dynamic Linking Library API.

We made an auxiliary tool based on Yolov3 frame, each time when we press a key, the weapon will automatically aim at the head of the enemy. All we need to do is click the mouse, and the gun or something else will hit the enemy automatically.

The motivation of this project is not to use it to cheat in the game, but to combine artificial intelligence with entertainment and improve the anti-cheating techniques in FPS game industry. Therefore, this project will only be used for academic research and discussion, and it is forbidden to use it for commercial sales and esports. Next details what and how CNN and yolov3 will be used in this auto-targeting tool.

# Methodology

This article mainly adopts literature research method and empirical research method. The former refers to the method of investigating ancient and modern papers and forming a scientific understanding of facts through the study of documents, while the latter means that empirical research methods mainly conduct quantitative analysis and speak based on data to make the research on social issues more precise and scientific [2].

First, we summarized the existing research on CNN and yolo to arrive at a framework suitable for this paper, and finally apply it to FPS games, especially CSGO.

# Related Works

## Introduction to FPS

According to different playing methods, games can be divided into many types, such as decryption games, shooting games, simulation games, and action games. Shooting games are one of the action games. According to different shooting angles, we can divide them into first-person and third-person shooting games. FPS refers to a shooting game conducted from the player's main perspective, that is, a shooting game in which the player uses his own perspective to experience the immersive experience of the protagonist.

## Introduction to CNN

Over the last decade, Convolutional Neural Networks have achieved ground-breaking breakthroughs in a variety of pattern recognition domains [2]. In addition, it is becoming more and more popular in recent years.

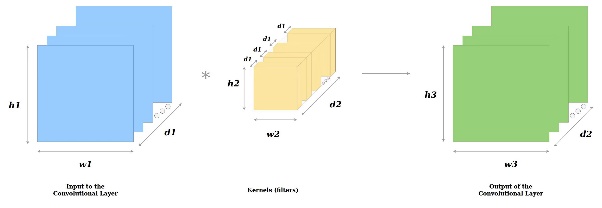
CNN is a feedforward neural network, the artificial neural its response within a portion around the cell coverage area, for large image processing have outstanding performance. CNN is a feedforward neural network, the artificial neural its response within a portion around the cell coverage area, for large image processing have outstanding performance [3].

An input layer, hidden layers, and an output layer comprise a convolutional neural network. The convolution procedure creates a feature map as the convolution kernel slides along the input matrix for the layer, which then contributes to the input of the following layer.

The activation function is very important for the artificial neural network model to learn and understand very complex and non-linear functions. They introduce the non-linear characteristics into our network. In the networks of neurons, after the input inputs are weighted and summed, a function is applied, and this function is called activation function. “ReLU” is one of them. In the yolo algorithm, we use the variance of “ReLU” function, that is the “Leaky ReLU” function.

## Introduction to Convolutional layers

In a CNN, the input is a tensor with a shape: (number of inputs) x (input height) x (input width) x (input depth) [4]. Why convolution? Assuming that the network receives an image with a width and height of 48\*48 pixels and a depth of 3 which RGB channel as input, to connect the input layer to one neuron. Then there should be 48\*48\*3 weight connections for this dataset.



1. Introduction to Convolutional Layers

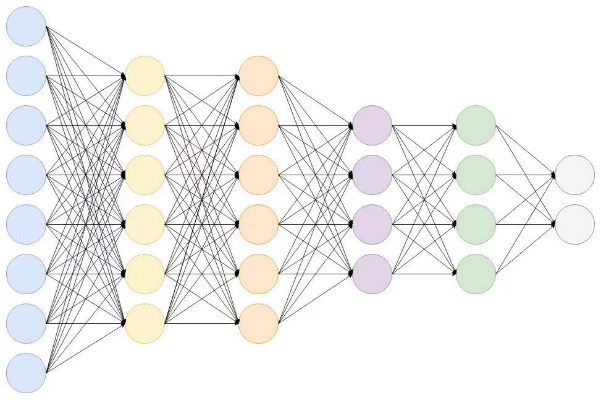
## Introduction to Pooling Layer

They progressively reduce data dimensionality by consolidating the output of neuron clusters at one player into a single neuron in the following layer. There are two types of pooling layers. One is the max pooling; another is the average pooling. The most common approach used in pooling is max pooling. In max pooling, for every position, the max value is taken [5].

## Introduction to Fully connected layers

The fully connected layer functions as a "classifier" in the entire convolutional neural network. If operations such as the convolutional layer, pooling layer, and activation function layer map the original data to the hidden layer feature space, the fully connected layer plays the role of mapping the learned "distributed feature representation" to the sample label space. In actual use, the fully connected layer can be realized by convolution operation.

The input of fully connected layers is the output of the final pooling or convolutional layer. To identify the photos, the flattened matrix is passed through a fully connected layer.

****

1. Introduction to Fully Connected Network

# IMAGE RECOGNITION WITH CNNS

The Gradient is a vector in mathematics that refers to the direction with the greatest changing value at one point.

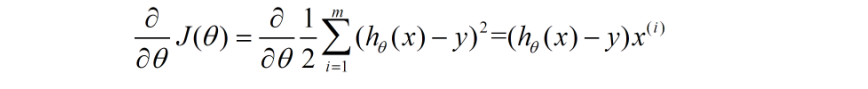
The gradient descent method is based on the following process:

1) First, assign the value of θ, this value can either be random or an all-zero vector.

2) Second, change the value of θ so that J (θ) decreases in the direction in which the gradient decreases [6].

The procedure is repeated for multiple small sets of instances from the training set until the average of the goal functions no longer decreases [4].

The Gradient reduction process is stated as follows [6]:



The backpropagation error is the sensitivity of the base of each neuron. It is defined as follows:

图片包含 文本

描述已自动生成

CNNs receive input for a specific neuron and then scale it using the neuron's delta.

## Distinguishing features

Traditional multilayer perceptron models were utilized for image recognition in the past. This type of design ignores the spatial structure of data. When pixels are separated by a large distance, this design treats them the same as when they are close together. As a result, full connectivity of neurons is wasteful for purposes dominated by spatially local input patterns. CNNs are variants of multilayer perceptron. CNNs, as opposed to MLPs, have the following distinguishing characteristics:

First, 3D volumes of neurons. A CNN layers have neurons of three dimensions: width, height, and depth. The neuron inside a convolutional layer is connected to the receptive field mentioned above.

Second, there is a local connection. CNNs enforce a pattern of local connection between neurons in neighbouring layers.

Third, there are shared weights. Each filter in a CNN is duplicated throughout the whole visual field. It can cut memory needs by reducing the amounts of free parameters learnt.

Forth, pooling. Ideally, the pooling method would maintain discriminative information while removing irrelevant visual elements [5]. This enables the CNN to be more robust to changes in their positions. These four characteristics enable CNNs to do greater generalization on visual challenges.

# Evaluation

The accuracy of the final model is based on a sub-part of the dataset set apart at the start, often called a test-set. K-fold cross-validation is applied in times methods, and there is another method called conformal prediction. Let's look at what they are in detail.

K-fold cross-validation: A cross-validation is a statistical approach for estimating the ability of machine learning models. It is largely used in applied machine learning to evaluate a machine learning model's skill on unseen data. That is, to use a small sample to assess how the model will perform in general when used to generate predictions on data that was not utilized during the model's training.

A restricted dataset is divided into k non-overlapping folds using the k-fold cross-validation process. Each of the k folds is given the option to be used as a held-back test set, while all other folds are utilized as a training dataset collectively. On the k hold-out test sets, a total of k models is fit and assessed, and the mean performance is provided.

The general procedure is as follows:

a) Shuffle the dataset randomly.

b) Split the dataset into k groups

c) For each unique group:

1. Take the group as a holdout or test data set

2. Take the remaining groups as a training data set

3. Fit a model on the training set and evaluate it on the test set

4. Retain the evaluation score and discard the model

d) Summarize the skill of the model using the sample of model evaluation scores [7].

Configuration of k: The essential configuration option for k-fold cross-validation is k, which specifies the number of folds into which a given dataset should be split. A poorly chosen value for k may result in an inaccurate representation of the model's competence. Common values include k=3, k=5, and k=10, with k=10 being the most commonly used in applied machine learning to assess models. This is because researches were conducted, and k=10 was discovered to give a suitable trade-off of cheap computing cost and minimal bias in an assessment of model performance. We may run a sensitivity analysis on various k values to see if the value makes sense for our dataset. Low values of k are expected to result in a noisy estimate of model performance, whereas high values of k are expected to result in a less noisy estimate of model performance.

# Yolo used In Our Program

## How YOLO used

First-person shooting games, FPS (First-person shooting game), strictly speaking, is a branch of action games, but like RTS games, due to its rapid popularity in the world, it has developed into A separate type. In fps games, the final thing is to quickly aim at the target and shoot, especially for head shooting. The headshot rate is the direct data showing the level of FPS games.

If we want to use artificial intelligence to achieve fps games, we need to use the YOLO algorithm to quickly locate the target, and then further position the head to complete the headshot and achieve our goal of FPS killer.

## How to Implement

Roughly speaking, target detection is: input pictures/videos, after processing, get: target location information, such as the coordinates of the upper left and lower right corners, the target's prediction category, and the target's prediction confidence.

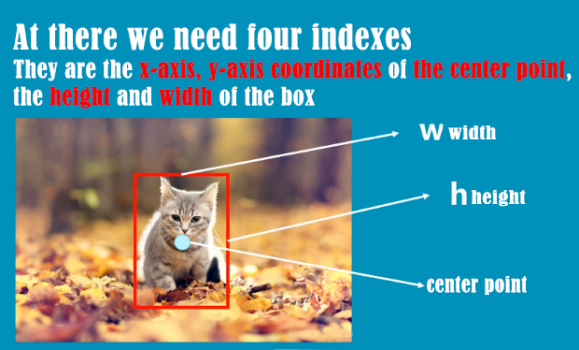
YOLOv3 detection is divided into two steps:

1. Determine the location of the detection object

2. Categorize the detection object (what is it)

That is, on the basis of identifying what the picture is, it is necessary to locate the position of the identified object and frame it.

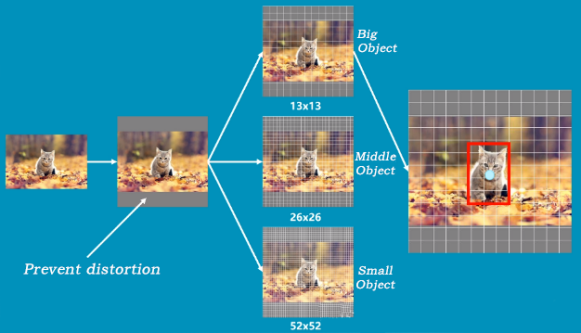
Let's first understand how our algorithm works in the follow picture.



1. YOLO Example Step 2 [8]

The red box in the figure is the bounding box finally obtained by detecting in YOLOv3, and the yellow box in the figure 4.2-3 below is also the bounding box.

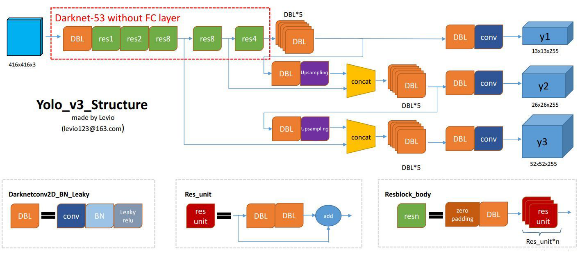
The process of YOLOv3 processing pictures is as follows.



1. YOLO Example Step 2 [8]

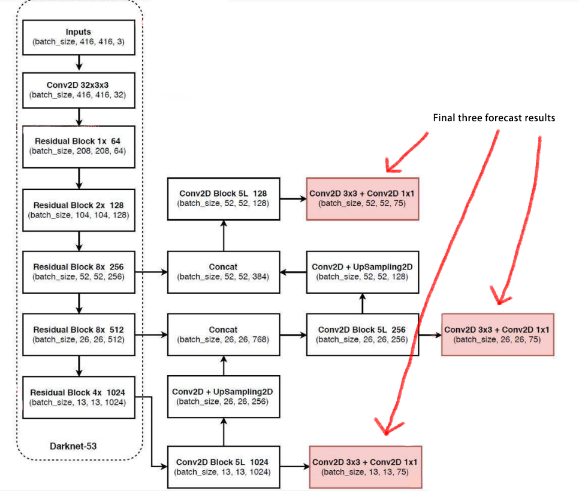
First, a picture is transferred to yolo, and yolo will convert it into a 384×384 grid, adding grey bars to prevent distortion, and then the picture will be divided into three grid pictures.

And the detail process will show as follow picture.



1. YOLO Example Step3 [10]

## Detailed Process



1. Detailed process [11]

**Step 1: Obtain prediction results**

1. YOLOv3 extracts multiple feature layers for target detection. A total of three feature layers are extracted. The three feature layers are located in different positions of the backbone feature extraction network darknet53. They are located in the middle layer, middle and lower layer, and bottom layer. The shapes of the three feature layers are respectively (52,52,256), (26,26,512), (13,13,1024), these three feature layers are used to stack and splice with other feature layers after upsampling [12].

2. The third feature layer (13,13,1024) performs 5 convolution processing. After processing, one part is used for convolution + upsampling, and the other part is used to output the corresponding prediction results (13,13,75), Conv2D 3×3 and Conv2D1×1 two convolutions play the role of channel adjustment, adjusted to the size required by the output [13].

3. After convolution + upsampling, the feature layer of (26, 26, 256) is obtained, and then it is spliced ​​with the feature layer (26, 26, 512) in the Darknet53 network, and the obtained shape is (26, 26, 768), and then convolution is performed 5 times After processing, one part is used for convolution up-sampling, and the other part is used to output the corresponding prediction results (26, 26, 75). Conv2D 3×3 and Conv2D1×1 are the same as above for channel adjustment

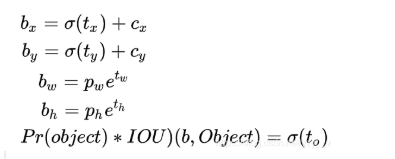
4. After that, splice the feature layer of convolution + upsampling in 3 with the feature layer of shape (52,52,256), and then convolve to obtain the feature layer of shape (52,52,128), and finally Conv2D 3×3 And Conv2D1×1 two convolutions to get (52,52,75) feature layer [14].

There are three red boxes in the last picture. The reason is that some objects are relatively large in the picture, so they are detected by 13×13. If the objects are relatively small in the picture, they will be classified as 52×52 for detection [15].

**Step 2: Decoding the prediction result**

Reason for decoding the prediction result: The prediction result (red box) does not correspond to the position of the final prediction box on the picture, and it needs to be decoded).

The prediction principle of yolov3 is to divide the entire image into 13x13, 26x26, 52x52 grids, and each network point is responsible for the detection of an area. The decoding process is to calculate the coordinates bx, by, width and height bw, bh of the last displayed bounding box, so that the position of the bounding box is obtained. The calculation process is shown in the figure (b-the abbreviation of bounding box) [16].



(cx,cy): The number of grids that differ from the upper left corner of the grid where the point is to the upper left corner.

(pw,ph): the side length of the a priori box

(tx,ty): the offset of the target center point relative to the upper left corner of the grid where the point is

(tw,th): predict the width and height of the border

σ: activation function, we used LeakyRuLU.

Step 3: Sort the predicted bounding box scores and non-maximum suppression screening.This step is to filter out the boxes with the highest probability. Take out each type of box with a score greater than a certain threshold and sort the scores. Then use frame position and score for non-maximum suppression. Finally, the bounding box with the highest probability can be obtained, which is the last displayed box. Follow we will use the rainbow six gaming screenshot to display the selection process [17].

First, we can see there are three border, and the score is 0.67, 0.77, 0.92 in incremental order.



1. Object Recognition step 1

and we will find the largest, obviously it’s 0.92 which is with red sign in the figure 4.3-3 below.



1. Object Recognition Step 2

In the last, we can see that the enemy is in the red border and that’s the object.



1. Object Recognition Step 3

# Testing

Now, we will continue a series of tests to prove the feasibility and accuracy of our project. Below we will test under different scenarios and maps.

And after we set many times test for each scenario and found the project are all works well and excellent. So we can say our project can work as our expectation. So for the test we draw two table for them:

1. Map Testing

|  |  |  |
| --- | --- | --- |
| **Map** | **Desert** | **City** |
| **Test time** | 100 | 100 |
| **Detection rate** | 97% | 99% |
| **Error rate** | 1% | 0% |

As we can see the project works very well and the detection rate is very high (only sometime there some obstacles or move too fast) and error rate is still very low.

1. Accuracy and Time in each Map

|  |  |  |  |
| --- | --- | --- | --- |
| **Orientation** | **head-up** | **Top-down** | **Bottom-up** |
| **Test time** | 100 | 100 | 100 |
| **Detection rate** | 99% | 100% | 98% |
| **Error rate** | 0% | 0% | 0% |

At there we can see the project works very excellent. At these 300 times test there’s so error happened, and the rate is very high.For top-down the enemy is easier to find so the rate can be reach 100% and for the other two cases perform also very well. These experiments can proved our project is fully meet expectations and highly successful.

# Conclusion

Our project FPS Killer takes YOLOv3 algorithm as the basic framework and applies convolution neural network (CNN) related technology to propose an object movement prediction technology. The project aims to explore the adaptive ability and practical application scenarios of mainstream deep learning models, try to give artificial intelligence projects more possibilities in daily life, and improve the anti-cheating technology in FPS game field [11].

In the whole project, CNN's related concepts and learning model evaluation methods are obtained through reliable existing literature, the calculation data used by YOLO algorithm has been reasonably screened and demonstrated, and the feasibility and stability of the project core code have been verified by many practices. At the same time, lots of related knowledge we learned in this semester's AI course has been deeply understood and effectively implemented.

As a test and expansion of the learning achievements of AI course, our project has provided background research and preliminary discussion on motion capture and prediction technology. As an example of computer science effectively learning from neuroscience, convolutional neural network has a wide range of applications. Although we generally associate CNN with image processing, in fact CNN can also process many Grid-like Data. Therefore, with the passage of time, the project may be extended to a broader and deeper research field, such as the processing of three-dimension data with time series [6].

As mentioned in the report, in the whole process of resource collection and project research, we have encountered many doubts and difficulties. However, our team members study hard and work together, continuously trained and improved the project model, and successfully obtained the final research results [18]. For each of us, the practical experience of this project is unforgettable and has important guiding significance.

##### Acknowledgment *(Heading 5)*

Special thanks to the IEEE for organizing such a well-run conference and for fostering a culture of innovation and progress. The meticulous efforts of the technical program committee, session chairs, and volunteers are truly appreciated. Their dedication and expertise ensured a smooth and productive conference.

I also wish to acknowledge my fellow researchers, collaborators, and mentors who have supported and guided me throughout my academic journey. Their invaluable inputs and encouragement have prepared me for the challenges and opportunities in this ever-evolving field

Lastly, I would like to recognize my family and friends for their unwavering love and support, which inspire me to continually strive for excellence. Their encouragement and patience have been instrumental in helping me succeed in both my personal and professional life.

##### References

[1] A. Liu, M. S. Khatun, H. Liu, and M. H. Miraz, “Lightweight blockchain of things (BCoT) architecture for enhanced security: a literature review,” in *2021 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA)*, IEEE, 2021, pp. 25–30. Accessed: Nov. 09, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9657112/

[2] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 international conference on engineering and technology (ICET)*, Ieee, 2017, pp. 1–6. Accessed: Nov. 09, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8308186/

[3] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, “Traffic-Sign Detection and Classification in the Wild,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2110–2118. Accessed: Oct. 07, 2023. [Online]. Available: https://openaccess.thecvf.com/content\_cvpr\_2016/html/Zhu\_Traffic-Sign\_Detection\_and\_CVPR\_2016\_paper.html

[4] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, “Deep convolution neural network for image recognition,” *Ecol. Inform.*, vol. 48, pp. 257–268, 2018.

[5] N. Akhtar and U. Ragavendran, “Interpretation of intelligence in CNN-pooling processes: a methodological survey,” *Neural Comput. Appl.*, vol. 32, no. 3, pp. 879–898, Feb. 2020, doi: 10.1007/s00521-019-04296-5.

[6] D. Gao *et al.*, “Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation.” arXiv, Sep. 20, 2023. Accessed: Oct. 08, 2023. [Online]. Available: http://arxiv.org/abs/2308.15363

[7] M. K. Hassan, A. I. El Desouky, S. M. Elghamrawy, and A. M. Sarhan, “A Hybrid Real-time remote monitoring framework with NB-WOA algorithm for patients with chronic diseases,” *Future Gener. Comput. Syst.*, vol. 93, pp. 77–95, Apr. 2019, doi: 10.1016/j.future.2018.10.021.

[8] W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang, “YOLO-face: a real-time face detector,” *Vis. Comput.*, vol. 37, no. 4, pp. 805–813, Apr. 2021, doi: 10.1007/s00371-020-01831-7.

[9] W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang, “YOLO-face: a real-time face detector,” *Vis. Comput.*, vol. 37, no. 4, pp. 805–813, Apr. 2021, doi: 10.1007/s00371-020-01831-7.

[10] “Remote Sensing | Free Full-Text | Improving YOLOv5 with Attention Mechanism for Detecting Boulders from Planetary Images.” Accessed: Oct. 07, 2023. [Online]. Available: https://www.mdpi.com/2072-4292/13/18/3776

[11] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks.” arXiv, Dec. 02, 2015. Accessed: Nov. 09, 2023. [Online]. Available: http://arxiv.org/abs/1511.08458

[12] M. Fazio, A. Celesti, F. G. Márquez, A. Glikson, and M. Villari, “Exploiting the FIWARE cloud platform to develop a remote patient monitoring system | IEEE Conference Publication | IEEE Xplore.” Accessed: Nov. 01, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7405526

[13] H. Mara and B. Bogacz, “Breaking the code on broken tablets: The learning challenge for annotated cuneiform script in normalized 2d and 3d datasets,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2019, pp. 148–153. Accessed: Nov. 09, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8978050/

[14] A. Angelopoulos, S. Bates, J. Malik, and M. I. Jordan, “Uncertainty Sets for Image Classifiers using Conformal Prediction.” arXiv, Sep. 03, 2022. Accessed: Nov. 09, 2023. [Online]. Available: http://arxiv.org/abs/2009.14193

[15] Mst. S. Khatun, A. Liu, and M. H. Miraz, “BCoT-Based Smart Manufacturing: An Enhanced Precise Measurement Management System,” in *Emerging Technologies in Computing*, vol. 463, M. H. Miraz, G. Southall, M. Ali, and A. Ware, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 463. , Cham: Springer Nature Switzerland, 2023, pp. 29–53. doi: 10.1007/978-3-031-25161-0\_3.

[16] 粤港澳大湾区数字经济研究院（福田）, “[PDF] DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction-论文阅读讨论-ReadPaper - 轻松读论文 | 专业翻译 | 一键引文 | 图表同屏,” ReadPaper论文阅读平台. Accessed: Oct. 20, 2023. [Online]. Available: https://readpaper.com/paper/4748055637330509825

[17] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, “Convolutional neural network (CNN) for image detection and recognition,” in *2018 first international conference on secure cyber computing and communication (ICSCCC)*, IEEE, 2018, pp. 278–282. Accessed: Nov. 09, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8703316/

[18] Y.-H. Kim, I.-K. Lim, J.-W. Lee, and J.-K. Lee, “Sensor Based Real-Time Remote Patient Monitoring System: A Study on Mobile DB Construction of Minimum Network Traffic in Use of HTML5 WebSQL,” *Procedia Eng.*, vol. 29, pp. 2382–2387, Jan. 2012, doi: 10.1016/j.proeng.2012.01.319.