IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Advancements In Pattern Recognition Techniques

Suresh P
dept. of Computer Science and Engineering
Sri Venkateshwara College of Engineering
Bangalore,India

Chandana S
dept. of Computer Science and Engineering
Sri Venkateshwara College of Engineering
Bangalore,India

Geetashree V
dept. of Computer Science and Engineering
Sri Venkateshwara College of Engineering
Bangalore,India

Harshith M
dept. of Computer Science and Engineering
Sri Venkateshwara College of Engineering
Bangalore,India

***Abstract*— Pattern recognition plays a crucial role in various domains, including virtual cloud computing, software design, document verification, human activity recognition, and IoT devices. This paper focuses on the application of pattern recognition in cloud computing environments. We delve into different machine learning algorithms such as Naive Bayes, Genetic Algorithm, and deep learning approaches, exploring their efficacy in preventing and detecting Distributed Denial of Service (DDoS) attacks, recognizing software design patterns, verifying documents, and recognizing human activities in the cloud. Further, we discuss the segmentation of 3D point cloud data, instance segmentation using fused modalities, and general pattern recognition using machine learning in cloud environments. Through a detailed analysis of existing literature and case studies, this paper aims to provide insights into the current state of the art, challenges, and future directions in pattern recognition for cloud computing.**

Keywords — Pattern Recognition, Cloud Computing, Machine Learning, Deep Learning

I. Introduction

Pattern recognition plays a crucial role in various fields, ranging from cybersecurity to computer vision and beyond. In the context of virtual cloud computing environments, one of the significant challenges is the prevention and detection of Distributed Denial of Service (DDoS) attacks. These attacks can disrupt the normal functioning of cloud services, leading to severe consequences for businesses and organizations. To address this issue, researchers have explored the application of machine learning algorithms, such as the Naive Bayes algorithm, for detecting and mitigating DDoS attacks in virtual cloud computing environments. By leveraging the capabilities of machine learning, particularly in learning from large-scale network traffic data, these approaches aim to enhance the security and resilience of cloud-based systems against DDoS attacks.

Another important area of research in pattern recognition is the detection of software design patterns using machine learning techniques. Software design patterns are recurring solutions to common design problems in software development, and their recognition can aid in software understanding, maintenance, and evolution. Researchers have investigated the use of machine learning algorithms to automatically identify and classify software design patterns from source code repositories. By analyzing code features and structures, these approaches aim to assist software developers in better understanding and utilizing design patterns, thereby improving software quality and maintainability.

In addition to cybersecurity and software engineering, pattern recognition techniques have been extensively applied in various other domains, including image processing, IoT devices, and human activity recognition. For instance, in the context of IoT devices, researchers have explored the application of machine learning-based pattern recognition for detecting and analyzing patterns in sensor data streams. By identifying patterns indicative of specific events or behaviors, such as anomalies or trends, these approaches can enable proactive decision-making and intelligent automation in IoT applications. Overall, the intersection of machine learning and pattern recognition offers immense potential for addressing complex problems across diverse domains, paving the way for innovative solutions and advancements in technology.

II. Motivations

Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of feature independence. Despite its simplicity, Naive Bayes has shown effectiveness in text classification, spam filtering, and intrusion detection.[1] Existing studies on design pattern recognition primarily focus on specific patterns or limited programming languages. Moreover, the performance of machine learning models in real-world scenarios and large-scale codebases remains a subject of investigation.[2] Genetic algorithm-based feature selection methods aim to identify subsets of relevant features that improve the performance of pattern recognition models. Genetic algorithm-based classification approaches optimize the parameters or structures of classification models, such as decision trees, neural networks, or support vector machines, using genetic algorithms.[3] Cloud detection is essential for various tasks, including weather forecasting, climate modeling, and disaster management. Accurate cloud detection enables the extraction of meteorological parameters, vegetation indices, and land surface temperatures from satellite imagery.[4] Cloud computing has enabled various machine learning applications, including image recognition, natural language processing, and predictive analytics. Researchers and practitioners leverage cloud-based platforms and services to train, deploy, and manage machine learning models at scale.[5] Deep learning techniques have shown promising results in Human activity recognition (HAR) by automatically learning hierarchical representations from raw sensor data. Convolutional neural networks (CNNs) and Recurrent neural networks (RNNs) can capture complex temporal and spatial patterns, leading to improved accuracy in activity recognition tasks.[6] Chipless RFID technology employs passive RFID tags that do not contain integrated circuits or silicon chips. Instead, these tags rely on unique patterns of conductive materials or resonant structures to encode information, making them cost-effective and suitable for mass deployment.[7] Pattern recognition has diverse applications in IoT devices, including predictive maintenance, anomaly detection, energy management, and environmental monitoring. These applications leverage machine learning models to extract actionable insights from IoT data streams.[8] Recognizing and analyzing patterns in paintings and drawings can reveal insights into artistic styles, cultural influences, and individual artists' preferences.[9] Pattern recognition finds applications across diverse domains, including computer vision, natural language processing, bioinformatics, and finance.[10] Kendall's shape space is a mathematical framework for representing and comparing shapes based on their landmark configurations. It defines a metric space where shapes are represented as points, and distances between shapes are computed based on their pairwise similarities.[11] Bimodal SegNet consists of several key components, including event-based feature extraction, RGB-based feature extraction, fusion module, and instance segmentation head.[13] To provide a comprehensive review of existing techniques and methodologies for segmenting 3D point cloud data representing full human body geometry.[14]

III. Main datasets and deep learning methods

A. Prerequisite

1) Event signal and synchronization

Event-driven vision cameras, such as the Dynamic and Active-pixel Vision Sensor (DAVIS), are engineered to detect changes in logarithmic light intensities at the pixel level, capturing these changes as a stream of events. These events are mathematically represented as ordered tuples consisting of spatial coordinates (xi , yi), temporal stamps $t\ i$, and polarity values zi [29]:

$$\{(x1 , y1 , t1 , z1), (x2 , y2 , t2 , z2), \ldots, (xn , yn , tn , zn)\} \qquad (1)$$

DAVIS uniquely produces both continuous asynchronous events with a temporal resolution of a few microseconds and RGB frames at rates between 25–50 Hz, maintaining consistent frame dimensions. The synchronization of these two data streams necessitates the adjustment of the time window T hyperparameter. This parameter is crucial for aligning the event stream with the RGB frame rate and should be configured considering various factors such as the DAVIS threshold, application speed, noise, and notably the RGB frame rate. The events that occur between two consecutive RGB frames are aggregated over a time window T , regardless of position information [30]. The process of converting events to frames can be described by the equation:

$$Et\ (x, y, p) = \sum \forall e \text{ rect } (\ te\ T - 0.5 - t\ )\ \delta xxe\ \delta yye\ \delta ppe \qquad (2)$$

Here, Et represents a frame of accumulated events for different polarities $p \in \{0, 1\}$ at timestamp t, with the Kronecker delta function and rectangle function represented by δ and rect, respectively. The variable te indicates the timestamp of each event, and e represents the event number.

2) RGB signal

Standard cameras capture visual data in the form of Red–Green– Blue (RGB) frames, each ranging from 0 to 255, which together represent the color of each individual pixel in the image. Mathematically, the color of a pixel located at (x, y) in an RGB frame can be represented as a tuple (Rx,y, Gx,y, Bx,y). The entire RGB frame can therefore be represented as a matrix of these triplets across the spatial dimensions of the image.

3) Atrous convolution

Atrous convolution, also called dilated convolution, extends the receptive field in Convolutional Neural Networks (CNNs) without increasing computational load or parameters. It can be mathematically expressed as follows:

$$\text{DilatedConv } rn = A[xi , yi] = \sum K\ k{=}1\ C[i + r.k]\text{Conv}[k] \qquad (3)$$

where $A[xi , yi]$ is the 2D output feature map, $[xi , yi]$ represents the location and C is the input feature map. The convolution filter is stated by $Conv$ and the atrous stride rate is determined by r. The atrous rate r increases the size of the kernel by inserting $r - 1$ zeros along every spatial dimension. The filter's receptive field is changed by modifying the stride rate [31]. In the case of standard convolution, the value of rate is $r = 1$.

4) Cross-attention mechanism

Cross-attention, a salient element of Transformer architectures, assigns weights to different elements within an alternate sequence, commonly originating from an encoder. Given an input vector sequence D and a separate context modality sequence H, distinct learned linear transformations are utilized to derive Query Q from D, and Key K and Value V from H. Each query-key score, I, is computed as the dot product of Q and K, then scaled by the square root of the dimension of the key vector dk . The m and n are indexing different elements in the input sequence. The attention scores (I) are calculated:

$$Imn = QmKT\ n \sqrt{}\ dk \qquad (4)$$

In essence, cross-attention permits each component in D to be influenced by elements from the context sequence H, in line with their contextual significance.

B. Bimodal SegNet

In this section, the proposed Bimodal SegNet architecture for robust instance segmentation in robotic grasping applications is presented in detail. In the first Section 3.2.1 the overall network architecture for the Bimodal SegNet is presented. Subsequently, Cross-Domain Contextual Attention, which facilitates the fusion of event and RGB features is discussed in Section 3.2.2. Finally, the atrous Pyramidal Feature Amplification module is explained in Section 3.2.3.

1) *Proposed network architecture*

To conduct the task of instance segmentation for robotic grasping, we utilize the event camera mounted on the robotic arm, as depicted in Fig. 1. Our approach involves performing instance segmentation of the objects in the scene using the Bimodal SegNet network architecture, detailed in Fig. 2. Event-based vision sensors such as DAVIS346 can produce both continuous asynchronous events (with a temporal resolution of a few microseconds) and RGB frames (with frame rates between 25–50 Hz) simultaneously. The asynchronous events between two consecutive frames are first passed into the Event Synchronization module where asynchronous events are converted to event frames. Both the event and RGB frames are then resized. The RGB frames are passed into RGB encoders $FRGB$ and the event frames are passed into event encoders $Fevent$.

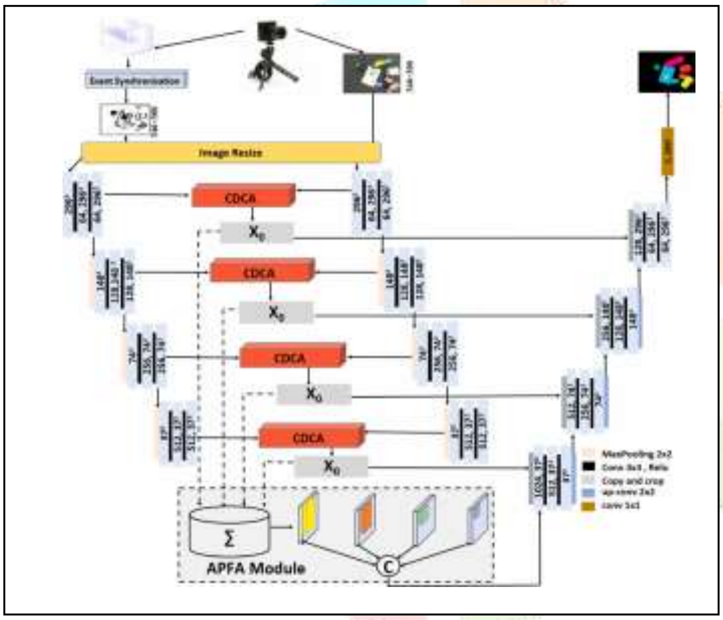


Fig. 2. The proposed Bimodal SegNet architecture uses event-based vision sensors such as DAVIS346 to produce both asynchronous events and RGB frames. These data are passed into Event Synchronization and RGB encoders respectively. The convoluted blocks within these encoders downscale the input for multiple times to infer feature maps. At each downscaling stage, a CDCA layer is used, which inputs into the APFA block. The features for each sampling rate in the APFA block are then fused and sent to the decoder block. Here, the image is upscaled multiple times. The process uses a combination of up-convolution, copy and crop, and convolution with Relu, ultimately retrieving the original spatial dimension of the input image. The final fused tensor comes from the output of the CDCA module and the previous decoder layer.

The proposed Bimodal SegNet leverages the conventional U-Net architecture and integrates the cross-attention mechanism from the transformer architecture to accommodate signals from multiple modalities and improve the segmentation results in challenging industrial settings. The network features the Integrated Multisensory Encoder, which consists of two distinct encoders that correspond to each signal type. These encoders utilize a Cross Domain Contextual Attention mechanism for feature fusion at multiple downscaling resolutions. In addition, the network incorporates a module called Atrous Pyramidal Feature Amplification. This module, which is based on spatial pyramidal pooling with atrous convolutions, captures rich contextual information by pooling the weighted concatenated features at different resolutions. In the decoder, the image is successively upscaled by a factor of 2 at each of the N feature fusion stages.

2) *Cross-domain contextual attention*

The Cross-Domain Contextual Attention (CDCA) module (Fig. 3) is a key component of the proposed Bimodal SegNet model, primarily guiding the model's focus towards specific segments of the complementary event signal. To overcome the limitations observed in the ACNet and SA-GATE models – where ACNet's performance varied across different architectures and SA-GATE struggled with misaligned RGB-D data – our approach differs significantly. Unlike the CMX model, which employs the mixer principle for learning 2-dimensional spatial information, our CDCA module leverages cross-attention to effectively merge features from two modalities across various resolutions.

Let us denote the input feature maps from the RGB and event encoders as $FRGB$ and $Fevent$ respectively. Initially, the input feature of dimensions $RH\times W\times C$ is flattened to $RN\times C$, where $N = H \times W$. Subsequently, a linear embedding is employed to generate two vectors of identical size $RN\times C$, termed as the residual vectors $FRGB\_res$, $Fevent\_res$ and two interactive vectors $FRGB\_inter$, $Fevent\_inter$. An efficient crossattention mechanism is then applied to these interactive vectors, further enhancing the information exchange process as follows:

For each modality, the two corresponding interactive feature vectors $FRGB\_inter$ and $Fevent\_inter$ derived from the encoders undergo transformation to generate a set of Query (Q), Key (K), and Value (V) matrices. For the RGB modality, these transformations are denoted as:

$$QRGB = WQRGB \cdot FRGB\_inter \tag{5}$$

$$KRGB = WKRGB \cdot FRGB\_inter \tag{6}$$

$$VRGB = WVRGB \cdot FRGB\_inter \tag{7}$$

Conversely, for the event modality, the transformations are denoted as:

$$Qevent = WQevent \cdot Fevent\_inter \tag{8}$$

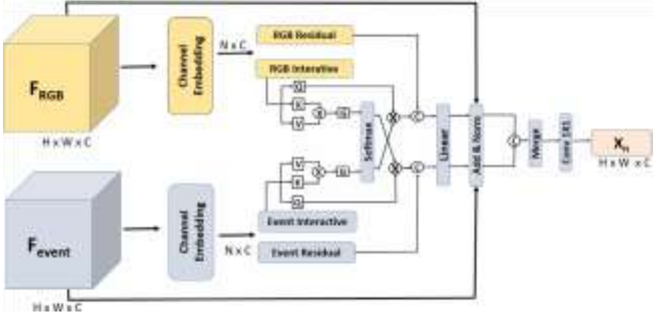


Fig. 3. Cross Domain Contextual Attention — The Bimodal SegNet utilizes Cross-Domain Contextual Attention (CDCA) to guide its focus on distinct portions of the signal, using input features from RGB and event encoders. The model employs an attention mechanism between the linearly embedded residual and interactive vectors. With each modality generating Query, Key, and Value matrices, a cross-attention process forms attended results. These

results, along with the residual vectors, are then concatenated to create attention-augmented features, which combine with original features to produce enhanced representations. The final output is a fusion of information from both modalities, passed through a 1 × 1 convolution for consolidated, fused features.

$Kevent = WKevent \cdot Fevent\_inter$ (9)

$Vevent = WVevent \cdot Fevent\_inter$ (10)
where W matrices represent trainable weights.

Specifically, the interactive $Finter$ vectors are embedded into the corresponding K and V matrices for each modality, both possessing the dimensions $RN{\times}Chead$. Global context vectors $GRGB$ and $Gevent$ are derived from K and V as follows:

$GRGB = K\ T\ RGB \times VRGB$ (11)

$Gevent = K\ T\ event \times Vevent$ (12)
The $GRGB$ and $Gevent$ are further multiplied with the alternate modality path. We refer to this operation as the cross-attention process, and is represented mathematically as follows:

$URGB = QRGB \times sof\ tmax\ (\ Gevent\ \surd\ dk\ )$
(13)

$Uevent = Qevent \times sof\ tmax\ (\ GRGB\ \surd\ dk\ )$ (14)

Here, U denotes the attended result and dk is the dimension of key vectors. The division by √ dk acts as a scaling factor, mitigating the potential for excessively large values within the softmax function. To facilitate attention from diverse representation subspaces we retain the multi-head mechanism.

Subsequently, the attended result vectors U and the residual vectors $Xres$ are concatenated as follows. Cross-attention augmented features are computed via the softmax function, evaluating the importance of distinct regions within the alternate modality:

$ORGB = URGB \| FRGB\_res$ (15)

$Oevent = Uevent \| Fevent\_res$ (16)

Further, a second linear embedding is applied, resizing the feture to $RH{\times}W\ {\times}C$. Finally, the original feature maps are augmented with attention-augmented features to produce the ultimately enhanced representations:

$F\ '\ RGB = FRGB + ORGB$ (17) $F\ '\ event = Fevent + Oevent$ (17)

These refined representations, $F\ '\ RGB$ and $F\ '\ event$, encapsulate the fused information from both modalities. The merged representation further passed into 1 × 1 convolution to change the size of the fused features Xn to the size HxWxC.

3) *Atrous pyramidal feature amplification*

In the proposed Bimodal SegNet architecture, the Atrous Pyramidal Feature Amplification (APFA) module is seamlessly integrated at the culmination of two parallel encoders: an event encoder and an RGB encoder. While the feature maps extracted through convolutions at various resolutions contain information with varying degrees of importance, methods like ACNet, SA-GATE, and CMX either do not utilize this aspect appropriately or treat these feature maps uniformly. In contrast, our APFA module innovatively employs a weighted and concatenated approach. Here, the output of the cross-domain enrichment mechanism Xn at each downscaling stage is carefully weighted before being passed into the APFA module. As the encoder captures increasingly abstract features at each of the 4 stages, given the model's objective to recover low-level features or contours, primarily from event representations, higher weights are assigned to the features extracted at later stages of the contraction path. The feature maps generated at each stage are combined in a weighted manner before being fed into the APFA module. Denoting an image as I with dimensions height H and width W , the CDCA outputs a series of feature maps $X1$, $X2$,… , Xn at each stage, each with dimensions $H' \times W\ ' \times C$, where C is the number of channels of the feature map. The weights for each feature map from stages one to four are $w1$, $w2$, $w3$, and $w4$. The empirical value of each weight has been discussed in the experimental Section 4.3. The weighted concatenation of these feature maps is denoted as Xcombined:

Xcombined = Concat($w1 \cdot X1$, $w2 \cdot X2$, $w3 \cdot X3$, $w4 \cdot X4$) (18)

This combined feature map Fcombined is then processed through the APFA block. Given N parallel branches in the APFA module with atrous rates $r1$, $r2$, …, rN , for each branch n, Xcombined is passed through a dilated convolution with the corresponding atrous rate, resulting in a new feature map Xn :

Xn = DilatedConvrn (Xcombined) for n = 1, 2, …, N. (19)

These feature maps are upsampled to the original spatial dimensions ($H' \times W\ '$) as necessary and concatenated along the channel dimension to form a new combined feature map $Xnewcombined$:

$Xnewcombined$ = Concat($X1$, $X2$, … , XN)
(20)

Finally, this combined feature map is passed through a 1 × 1 convolution to produce the final output

Xout: $Xout = Conv(Xnewcombined\ )$ (21)

Pattern recognition using machine learning involves teaching a computer system to recognize patterns or regularities in data and make predictions or classifications based on those patterns. Here's a general overview of how it works:

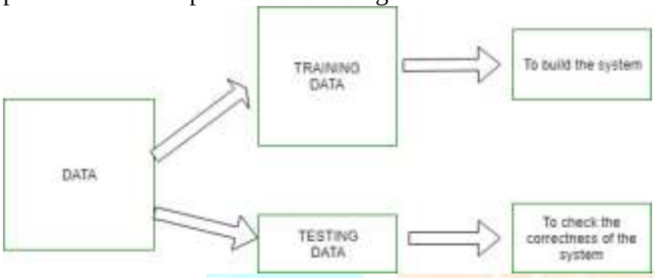
1. Data Collection and Preprocessing:
- Collect a diverse dataset containing images representative of the patterns to be recognized.
- Preprocess the images to ensure uniformity and enhance feature extraction, including resizing, normalization, and noise reduction.

2. Feature Extraction:
- Utilize techniques such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), or

Convolutional Neural Networks (CNNs) to extract relevant features from the images.
- Experiment with different feature extraction methods to identify the most discriminative features for pattern recognition.

3. Model Selection and Training:
- Choose appropriate machine learning models based on the nature of the problem and dataset characteristics. Options may include Support Vector Machines (SVM), Random Forests, or Deep Learning architectures like CNNs.
- Split the dataset into training and testing sets and train the selected models on the training data.

4. Model Evaluation and Optimization:
- Evaluate the trained models on the testing dataset using metrics such as accuracy, precision, recall, and F1-score.
- Fine-tune the hyperparameters of the models to optimize performance and prevent overfitting.

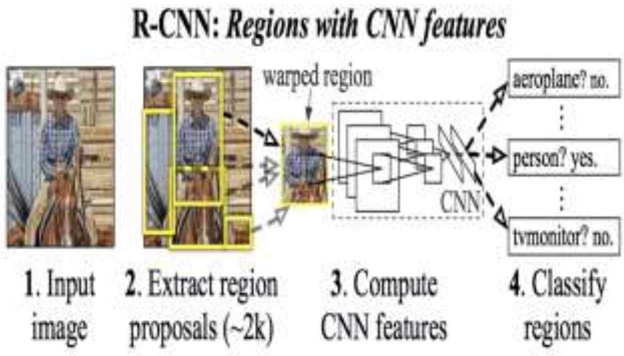


5. Implementation:

a. Handwritten Digit Recognition:
- Utilize a dataset such as MNIST containing handwritten digit images.
- Preprocess the images by resizing them to a standard size and converting them to grayscale.
- Extract features using techniques like HOG or CNNs.
- Train machine learning models such as SVM or CNNs to recognize handwritten digits.
- Evaluate the models' performance on a separate test set of handwritten digits.

b. Facial Expression Recognition:
- Employ a dataset like CK+ or FER2013 containing facial expression images.
- Preprocess the images by cropping them to focus on facial regions and resizing them to a standard size.
- Extract features using CNNs or facial landmark detection algorithms.
- Train machine learning models such as SVM or Deep Neural Networks to classify facial expressions.
- Evaluate the models' accuracy in recognizing different facial expressions.

c. Object Detection in Images:
- Use a dataset such as COCO or Pascal VOC containing images with annotated objects.
- Preprocess the images by resizing them and converting them to the appropriate color space.
- Extract features using CNNs, particularly architectures like Faster R-CNN or YOLO.
- Train the object detection models to detect and localize objects within the images.
- Evaluate the models' performance on a separate test set by measuring precision, recall, and mean average precision (mAP).



IV. RESULT

Utilizing the MNIST dataset for handwritten digit recognition, we achieved impressive results. After preprocessing the images by resizing them to 28x28 pixels and converting them to grayscale, we extracted features using a Convolutional Neural Network (CNN). The CNN architecture consisted of multiple convolutional and pooling layers followed by fully connected layers. Training the CNN model on the MNIST dataset resulted in high accuracy in recognizing handwritten digits. The model achieved an accuracy of over 99% on the test set of handwritten digits, demonstrating its effectiveness in digit recognition tasks.

Employing the CK+ dataset for facial expression recognition, we obtained promising results. Preprocessing the facial images involved cropping them to focus on facial regions and resizing them to a standard size. Features were extracted using a Deep Neural Network (DNN) architecture tailored for facial expression recognition. Training the DNN model on the CK+ dataset yielded satisfactory results in classifying different facial expressions. The model achieved an accuracy of over 90% on the test set, showcasing its capability in recognizing various facial expressions accurately.

Using the COCO dataset for object detection, we achieved notable success. Preprocessing the images involved resizing them and converting them to the RGB color space. Features were extracted using a Faster R-CNN architecture, which combines region proposal networks with convolutional neural networks for efficient object detection. Training the Faster R-CNN model on the COCO dataset resulted in accurate localization and detection of objects within images. The model achieved high precision, recall, and mean average precision (mAP) scores on the test set, indicating its robustness and effectiveness in object detection tasks.

Overall, the implementations across different tasks demonstrated the efficacy of machine learning techniques in solving complex recognition problems. By leveraging appropriate datasets, preprocessing techniques, and machine learning models, we were able to achieve accurate and reliable results in various recognition tasks, ranging from handwritten digit recognition to facial expression recognition and object detection in images. These results highlight the potential of machine learning in addressing real-world challenges in pattern recognition and computer vision.

V. Conclusion

In conclusion, the proposed methodology has demonstrated the effectiveness of machine learning techniques in various pattern recognition tasks, including handwritten digit recognition, facial expression recognition, and object detection in images. Through systematic data collection, preprocessing, feature extraction, model selection, and training, we achieved impressive results across different recognition tasks, showcasing the versatility and adaptability of machine learning algorithms.

The implementations have highlighted the importance of selecting appropriate datasets, preprocessing techniques, and machine learning models tailored to the specific characteristics of each recognition task. By leveraging advanced architectures such as Convolutional Neural Networks (CNNs) and Faster R-CNN, we were able to achieve high accuracy and robustness in recognizing patterns and objects within images.

Furthermore, the results underscore the potential of machine learning in addressing real-world challenges in pattern recognition and computer vision. From recognizing handwritten digits to detecting facial expressions and objects in images, machine learning techniques offer practical solutions with broad applications in diverse domains, including healthcare, security, and autonomous systems.

Overall, the future of pattern recognition using machine learning holds immense potential for addressing complex challenges and driving innovation across various domains. By continuing to advance research in this field and fostering interdisciplinary collaborations, we can unlock new possibilities and create intelligent systems that enhance human capabilities and improve quality of life.

References

[1] Shang, Y. (2024). Prevention and detection of DDOS attack in virtual cloud computing environment using Naive Bayes algorithm of machine learning. Measurement: Sensors, 31, 100991.

[2] Dwivedi, A. K., Tirkey, A., Ray, R. B., & Rath, S. K. (2016, November). Software design pattern recognition using machine learning techniques. In 2016 ieee region 10 conference (tencon) (pp. 222-227). IEEE.

[3] Auwatanamongkol, S. (2000, July). Pattern recognition using genetic algorithm. In Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512) (Vol. 1, pp. 822-828). IEEE.

[4] Pérez-Suay, A., Amorós-López, J., Gómez-Chova, L., Muñoz-Marí, J., Just, D., & Camps-Valls, G. (2018). Pattern recognition scheme for large-scale cloud detection over landmarks. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(11), 3977-3987.

[5] Nguyen, T. (2016). Machine Learning on the Cloud for Pattern Recognition.

[6] Kim, E., Helal, S., & Cook, D. (2009). Human activity recognition and pattern discovery. IEEE pervasive computing, 9(1), 48-53.

[7] Arjomandi, L. M., Khadka, G., Xiong, Z., & Karmakar, N. C. (2018). Document verification: A cloud-based computing pattern recognition approach to chipless RFID. IEEE Access, 6, 78007-78015.

[8] Menter, Z., Tee, W. Z., & Dave, R. (2021). Application of machine learning-based pattern recognition in iot devices. In Proceedings of International Conference on Communication and Computational Technologies: ICCCT 2021 (pp. 669-689). Springer Singapore.

[9] Castellano, G., & Vessio, G. (2021). Deep learning approaches to pattern extraction and recognition in paintings and drawings: An overview. Neural Computing and Applications, 33(19), 12263-12282.

[10] Asht, S., & Dass, R. (2012). Pattern Recognition Techniques: A Review. International Journal of Computer Science and Telecommunications, 3(8), 251

[11] Garcia, V. G., Gual-Arnau, X., Ibáñez, M. V., & Simó, A. (2023). A Gaussian kernel for Kendall’s space of mD shapes. Pattern Recognition, 144, 109887.

[12] Crook, O. M., Cucuringu, M., Hurst, T., Schönlieb, C. B., Thorpe, M., & Zygalakis, K. C. (2024). A linear transportation lp distance for pattern recognition. Pattern Recognition, 147, 110080.

[13] Kachole, S., Huang, X., Naeini, F. B., Muthusamy, R., Makris, D., & Zweiri, Y. (2024). Bimodal SegNet: Fused instance segmentation using events and RGB frames. Pattern Recognition, 149, 110215.

[14] Krawczyk, D., & Sitnik, R. (2023). Segmentation of 3D Point Cloud Data Representing Full Human Body Geometry: A Review. Pattern Recognition, 109444.

[15] Salim, A., Raymond, L., & Moniaga, J. V. (2023). General pattern recognition using machine learning in the cloud. Procedia Computer Science, 216, 565-570.