LITERATURE SURVEY

PROJECT TITLE - Sentinel: Intelligent Multi Camera Face Detection, Recognition and Tracking System for Advanced Video Surveillance

Roll number	Student Name
20201CAI0128	Rishi Ragav V
20201CAI0107	Israr Ahmed
20201CAI0090	Mohd Faizan Usman Sait
20201CAI0117	Rakshith M B

Under the Supervision of **Mr. S K Jamil Ahmed**, Assistant Professor, School of Computer Science and Engineering, Presidency University

Probabilistic recognition of human faces from video

Shaohua Zhou,Volker Krueger, and Rama Chellappa

The paper explores probabilistic video analysis for face recognition, building on the CONDENSATION algorithm (particle filter) and a time series state space model. In contrast to traditional still-to-still face recognition, the study focuses on the still-to-video scenario, addressing challenges like poor video quality and pose variations. The proposed tracking-and-recognition approach integrates temporal information, simultaneously resolving uncertainties in tracking and recognition. The model incorporates a motion equation, identity equation, and observation equation to characterize kinematics and identity evolution in the probe video. Using sequential importance sampling (SIS), the joint posterior distribution is estimated, facilitating computational efficiency in still-to-video recognition. The methodology is extended to video-to-video recognition by generalizing still templates to video sequences. Experiments demonstrate the effectiveness of the proposed approach, particularly in handling pose variations. The paper concludes by discussing the adaptability of the model to various recognition algorithms and transformations. Overall, it offers a unified probabilistic framework for face recognition in dynamic video sequences.

Terms used in this paper:

1. **CONDENSATION Algorithm:** Also known as the particle filter (PF), it's a numerical approximation method for the posterior distribution of a motion vector in tracking applications.
2. **Time Series State Space Model:** A mathematical framework used to represent and analyse time-dependent systems, where the evolution of variables is modelled over time.
3. **Sequential Importance Sampling (SIS):** A technique for estimating the joint posterior distribution of variables over time, often used in Bayesian filtering.
4. **Still-to-Still Recognition:** Traditional face recognition scenarios where both the gallery and probe sets consist of still facial images.
5. **Still-to-Video Scenario:** The gallery consists of still facial templates, while the probe set consists of video sequences containing the facial region.
6. **Video-to-Video Recognition:** Extending the recognition to video sequences, where exemplars and their prior probabilities are learned from gallery videos.
7. **Exemplar-Based Learning:** Learning from specific examples (exemplars) to improve recognition, allowing for adaptation to variations in poses and other factors.
8. **Kinematics:** The study of motion, here referring to the dynamic behaviour of the tracking motion vector.
9. **Likelihood Measurement:** The measure of how well a model explains observed data, often used in Bayesian inference.

10. **Pose Variations:** Changes in the orientation or position of the face, a common challenge in face recognition.
11. **Image Representations and Transformations:** Different ways of representing and transforming facial images, essential for recognition algorithms.
12. **Mixture Density:** A statistical model where the overall probability distribution is a combination (mixture) of several component distributions.
13. **Monte Carlo Method:** A statistical technique that uses random sampling to obtain numerical results.
14. **Marginal Distribution:** The distribution of one or more variables in a subset of a larger probability distribution
15. **Joint Distribution:** The probability distribution of multiple variables considered together.
16. **Computational Load:** The number of computational resources required by an algorithm.
17. **Pose Variations:** Changes in the orientation or position of an object, often referring to facial poses in face recognition.

TEINet: Towards an Efficient Architecture for Video Recognition

Zhaoyang Liu,1∗ Donghao Luo,2∗ Yabiao Wang,2 Limin Wang,1† Ying Tai,2
Chengjie Wang,2 Jilin Li,2 Feiyue Huang,2 Tong Lu1
1State Key Lab for Novel Software Technology, Nanjing University, China
2Youtu Lab, Tencent

The paper addresses efficiency concerns in video action recognition, proposing a Temporal Enhancement-and-Interaction (TEI) Module for integration into existing 2D Convolutional Neural Networks (CNNs). The TEI Module introduces a Motion Enhanced Module (MEM) for feature enhancement and a Temporal Interaction Module (TIM) for capturing temporal context. This modular approach flexibly captures temporal structure while maintaining computational efficiency. Extensive experiments on benchmarks, including Kinetics and Something-Something, validate the effectiveness of the proposed TEINet architecture. TEINet achieves state-of-the-art performance on Something-Something and comparable results to 3D CNNs on Kinetics with lower computational costs, demonstrating its generalization ability.

Terms used in this paper:

1. **Temporal Enhancement-and-Interaction (TEI) Module:** A proposed module for integrating temporal features into 2D CNNs, consisting of a Motion Enhanced Module (MEM) and a Temporal Interaction Module (TIM).
2. **Motion Enhanced Module (MEM):** A component of the TEI Module designed to enhance motion-related features while suppressing irrelevant information.
3. **Temporal Interaction Module (TIM):** A component of the TEI Module that supplements temporal contextual information in a channel-wise manner.
4. **2D CNNs:** Two-dimensional Convolutional Neural Networks, widely used in computer vision tasks.
5. **Spatial Domain:** The area or space in the context of video frames.
6. **Temporal Order Information:** The sequential arrangement of frames over time in videos.
7. **TSN (Temporal Segment Networks):** An efficient action recognition method that aggregates temporal information at the final classifier layer.
8. **StNet:** A 2D CNN-based architecture developed for efficient action recognition, capturing temporal information earlier.
9. **TSM (Temporal Shift Module):** A 2D CNN-based architecture designed for efficient temporal modelling.
10. **3D CNNs:** Three-dimensional Convolutional Neural Networks, used for directly learning spatiotemporal features from RGB frames in videos.
11. **Spatiotemporal Features:** Features that capture information both in space and time.
12. **Computational Cost:** The amount of computational resources required by an algorithm or model.
13. **Enhance-and-Interact:** The proposed design philosophy that decouples temporal modelling into two stages: enhancing discriminative features and capturing their temporal interaction.
14. **Channel-Level Correlation:** The correlation between channels in a neural network, representing different features.

15. **Discriminative Features:** Features that are distinctive and informative for the task at hand.
16. **Local Temporal Variations:** Changes in visual features occurring over short time intervals.
17. **ResNets (Residual Networks):** A type of deep neural network architecture known for its use of residual blocks.
18. **Generalization Ability:** The ability of a model to perform well on data it hasn't seen during training.
19. **Fine-Tuning:** Adjusting a pre-trained model on a new dataset to adapt it for a specific task.
20. **Fine-Tuning:** Adjusting a pre-trained model on a new dataset to adapt it for a specific task.

Design and Implementation of Object Detection, Tracking, Counting and Classification Algorithms using Artificial Intelligence for Automated Video Surveillance Applications

Mohana, Research Scholar, Electronics & Communication Engg.
Dr. H. V. Ravish Aradhya, Research Advisor

This research focuses on enhancing video surveillance through efficient object detection, tracking, counting, and classification. The implementation involves Matlab, DSP, FPGA (Zynq XC7Z020), and Artificial Intelligence (AI) methods, utilizing Convolutional Neural Networks (CNN), YOLO, SSD, and Modified Background Subtraction. AI combines SSD and Mobile Nets for efficient detection and tracking. FPGA and DSP implementations optimize speed and memory usage. The research aims to automate surveillance, reduce human intervention, and address security challenges. Results demonstrate the effectiveness of the proposed methods, with AI achieving 85.97% accuracy in image classification. Ongoing work explores NVIDA GPU technology for further improvements.

Terms used in this paper

1. **Video Surveillance:** The systematic monitoring of activities, behaviours, or events using video cameras.
2. **Object Detection:** The process of locating and identifying objects within a visual scene, often involving the use of algorithms and computer vision techniques.
3. **Tracking:** Continuous monitoring and analysis of the movement or changes in position of objects over time.
4. **Classification:** Categorizing objects or events into predefined classes or groups based on certain characteristics or features.
5. **Matlab:** A high-performance numerical computing environment and programming language used for algorithm development, data analysis, and visualization.
6. **DSP (Digital Signal Processing):** The manipulation and analysis of signals represented in a digital form, commonly used in audio and image processing.
7. **FPGA (Field-Programmable Gate Array):** A programmable integrated circuit that can be configured by a user after manufacturing, often employed for hardware acceleration in computing.
8. **Artificial Intelligence (AI):** The development of computer systems that can perform tasks that typically require human intelligence, such as problem-solving, learning, and decision-making.
9. **Convolutional Neural Network (CNN):** A type of deep neural network designed for image processing and pattern recognition, using convolutional layers for feature extraction.

10. **YOLO (You Only Look Once):** An object detection algorithm that processes the entire image in a single forward pass, known for its real-time performance.
11. **SSD (Single Shot Detector):** A type of object detection algorithm that predicts bounding boxes and class scores for multiple objects in a single pass.
12. **Modified Background Subtraction:** An algorithmic technique for identifying foreground objects in a video sequence by comparing each frame to a reference background.
13. **Image Classification:** Assigning a label or category to an image based on its visual content, often achieved through machine learning models.
14. **Autonomous Navigation:** The ability of a system, often a robot or vehicle, to navigate and make decisions without direct human intervention.
15. **Real-time Detection:** Immediate identification of objects or events as they occur, with minimal delay.
16. **Anchor Box Technique:** A method in object detection algorithms, like YOLO, involving the use of predefined bounding boxes to improve accuracy.
17. **Semantic Information:** Meaningful information that conveys the intended understanding or interpretation.
18. **Threat Detection:** Identification and assessment of potential dangers or risks in each environment.

A System for Video Surveillance and Monitoring

Robert T. Collins, Alan J. Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt1 and Lambert Wixson

The Carnegie Mellon University (CMU) research under the DARPA Video Surveillance and Monitoring (VSAM) project focuses on cooperative multi-sensor surveillance for battlefield awareness. The goal is to enhance situational awareness by developing automated video understanding technology. This technology enables a single human operator to monitor activities over a complex area using a distributed network of active video sensors. Applications include battlefield awareness, perimeter security, peace treaty monitoring, refugee movement surveillance, embassy, and airport security, and tracking suspicious activities such as drug or terrorist hideouts. Automated video surveillance addresses challenges in the commercial sector, where mounting cameras is affordable, but human resources for continuous monitoring are expensive. The research emphasizes real-time monitoring and analysis of video surveillance data to alert security officers promptly. The VSAM technology involves parsing people and vehicles from raw video, determining their geolocations, and inserting them into a dynamic scene visualization. It includes robust routines for detecting and tracking moving objects, classifying objects into semantic categories, and determining geolocations using stereo vision or terrain models. The system tasks multiple sensors with variable pan, tilt, and zoom to cooperatively track objects through the scene. The VSAM testbed system consists of multiple sensors distributed across the CMU campus, linked to a central operator control unit (OCU). The OCU integrates information from sensors, 3D geometric site models, and databases of known objects. The system employs a graphical user interface (GUI) for a single human operator to effectively monitor the area. Sensor processing units (SPUs) analyse video imagery and transmit symbolic data packets to the OCU. The communication architecture uses the Carnegie Mellon University Packet Architecture (CMUPA) and Distributed Interactive Simulation (DIS) protocols. The system aims to demonstrate the effectiveness of a single operator monitoring a significant area through an interactive GUI. The report is organized into sections describing the VSAM testbed system, video understanding algorithms, geospatial site models, coordination of multiple cameras, and achievements through yearly demos.

Terms used in this paper

1. **Semantic Categories:** Classification of detected objects into meaningful categories (e.g., human, human group, car, truck).
2. **Stereo Vision:** The ability to perceive depth by processing visual information from two or more cameras or sensors.

3. **Symbolic Data Packets:** Data packets containing symbolic representations of detected activities.
4. **Operator Control Unit (OCU):** Central unit responsible for integrating symbolic object trajectory information, a 3D geometric site model, and presenting results to the user.
5. **Graphical User Interface (GUI):** A visual interface that allows users to interact with electronic devices through graphical elements.
6. **Sensor Processing Units (SPUs):** Intelligent filters between cameras and the VSAM network, responsible for analysing video imagery and transmitting symbolic information to the OCU.
7. **Distributed Interactive Simulation (DIS):** A protocol for communication between entities in a distributed simulation.
8. **Synthetic View of the Environment:** An artificial representation of a scene or environment created using data from multiple sensors.
9. **Wide Baseline Stereo:** Stereo vision using cameras with a significant separation, allowing for the triangulation of objects at a distance.
10. **Perimeter Security:** Security measures designed to protect the outer boundaries of a facility or area.
11. **Object Hypotheses:** Assumptions or predictions about the presence and characteristics of objects in the surveillance scene.

Real-Time Flying Object Detection with YOLOv8

Dillon Reis*, Jordan Kupec, Jacqueline Hong, Ahmad Daoudi
Georgia Institute of Technology

The paper introduces a comprehensive approach to real-time detection of flying objects, motivated by recent events showcasing the malicious use of drones and the increasing prevalence of drone technology. The primary aim is to provide a generalized model for flying object detection suitable for transfer learning and further research, along with a refined model ready for immediate implementation. Notably, drones' unique characteristics, including their small size, manoeuvrability, and low electromagnetic signature, pose challenges to traditional detection methods, prompting the need for a visual detector.

The proposed methodology involves training an initial (generalized) model on a diverse dataset containing 40 different classes of flying objects. This forces the model to extract abstract feature representations, addressing challenges such as occlusion, small spatial sizes, rotations, and clustered backgrounds. Subsequently, transfer learning is employed on a second dataset, more representative of real-world scenarios, to generate a refined model. The YOLOv8 single-shot detector, chosen for its assumed state-of-the-art status, is implemented, emphasizing the trade-off between inference speed and mean Average Precision (mAP).

Two datasets are utilized: the first with 15,064 images of various flying objects, and the second focusing on challenges like distance, comprising 11,998 images. Evaluation metrics include mAP50-95 and average inference speed on 1080p videos. The YOLOv8 architecture, though lacking an official paper, is selected for its perceived superiority in mAP and inference speed on the COCO dataset, particularly excelling in detecting aerial objects. The training process involves transfer learning with pre-trained weights from COCO, and a greedy model selection approach, optimizing hyperparameters for model size and performance trade-offs. The results demonstrate the effectiveness of the proposed approach, with the refined model achieving an improved mAP50-95 of 0.835 while maintaining an impressive average inference speed. Overall, the paper provides a comprehensive solution to the challenging task of real-time flying object detection, addressing the nuances of diverse classes and real-world scenarios.

Terms used in this paper

1. **Flying Objects:** Refers to objects capable of flight, such as drones, airplanes, helicopters, and birds.
2. **Transfer Learning:** The technique of training a model on one task and applying its knowledge to a different, but related, task.
3. **Real-time Object Detection:** The ability to identify and locate objects within a video or image stream in real-time.

4. **YOLOv8:** Stands for "You Only Look One Level 8," a specific version of the YOLO (You Only Look Once) object detection model.
5. **mAP (Mean Average Precision):** An evaluation metric for object detection algorithms that calculates the average precision over all classes.
6. **Inference Speed:** The speed at which a model can make predictions or detections, particularly relevant for real-time applications.
7. **Occlusion:** The obstruction of one object by another in the field of view, making detection challenging.
8. **Spatial Sizes/Aspect Ratios:** The physical dimensions and proportions of objects, which can vary widely and present challenges for detection.
9. **Single-shot Detector:** A type of object detection model designed to make predictions in a single pass through the network.
10. **COOC Dataset:** Common Objects in Context dataset, widely used for benchmarking object detection models.
11. **Hyperparameters:** Parameters that are set before the training process and are not learned from the data.
12. **Greedy Model Selection:** A method of choosing the best-performing model by optimizing specific criteria, often making locally optimal choices at each step.
13. **IoU (Intersection over Union):** A measure of the overlap between the predicted bounding box and the ground truth bounding box.
14. **Roboflow:** A platform for managing and augmenting datasets for machine learning.
15. **UAVs (Unmanned Aerial Vehicles):** Another term for drones, emphasizing their autonomous and unmanned nature.
16. **COCO Dataset:** Common Objects in Context dataset, commonly used for training and evaluating object detection models.

Summary of YOLOv8: Advancements in Real-Time Object Detection

Introduction

You Only Look Once version 8 (YOLOv8) represents a significant milestone in the field of computer vision, particularly in the domain of real-time object detection. Building upon the successes and lessons learned from its predecessors, YOLOv8 introduces novel architectural enhancements to strike a balance between speed and accuracy. This article delves into the key components and improvements that characterize YOLOv8, exploring its design principles, advancements, and applications.

Background

The YOLO series, known for its groundbreaking approach to object detection, introduced the concept of processing an entire image in one forward pass through the neural network. This design drastically improved real-time object detection compared to traditional two-step methods. YOLOv8, an evolution of this concept, aims to refine and optimize the trade-off between detection precision and computational efficiency.

Architecture Overview

YOLOv8's architecture is built upon the Darknet neural network framework, specifically utilizing the CSPDarknet53 backbone. This choice ensures a powerful feature extraction process, allowing the model to understand intricate patterns and representations within the input data.

One notable addition to YOLOv8 is the integration of PANet, or Path Aggregation Network. PANet enhances information flow across different scales, addressing challenges related to recognizing objects of varying sizes within an image. This feature contributes to improved contextual understanding and object localization.

Multi-Scale Detection

One of YOLOv8's strengths lies in its multi-scale approach. By processing multiple scales concurrently, the model becomes adept at detecting objects of different sizes within the same image. This is crucial for scenarios where objects may vary significantly in scale, such as in surveillance footage or autonomous driving applications.

Detailed explanation of multi-scale detection:

1. **Object Size Variability:** In real-world images, objects can appear in various sizes due to factors like distance from the camera, perspective, and the inherent size variation of objects in the scene. For effective object detection, a model needs to be capable of identifying and localizing objects regardless of their size.
2. **Hierarchical Feature Representation:** Multi-scale detection involves processing an image at different resolutions or levels of detail. This is typically achieved through the use of feature pyramids or hierarchical feature representations. Lower-resolution features may capture larger, more global context, while higher-resolution features focus on finer details.

3. **Adaptive Receptive Fields:** Different objects may require different receptive fields (the region of the input space that a neuron is sensitive to) for accurate detection. Multi-scale detection ensures that the model has adaptive receptive fields, enabling it to capture both the overall context and specific details necessary for identifying objects.
4. **Handling Small and Large Objects:** Some object detection models might struggle with detecting small objects if they are optimized for larger ones, and vice versa. Multi-scale detection addresses this issue by considering features at different levels of granularity, making the model more robust across a broad spectrum of object sizes.
5. **Improving Localization Accuracy:** Multi-scale detection contributes to more accurate object localization. By considering features at different scales, the model can refine its predictions based on contextual information. This is crucial for precise localization, especially when dealing with objects that may be partially obscured or located in complex scenes.
6. **Application in YOLOv8:** YOLOv8 employs a multi-scale approach by processing the input image at various resolutions simultaneously. This is facilitated by the use of different layers in the network that capture features at different scales. The integration of PANet (Path Aggregation Network) in YOLOv8 further enhances multi-scale detection by aggregating information across different paths within the network.

CSPDarknet53 Backbone

The CSPDarknet53 backbone is a critical element of YOLOv8's architecture. Based on the concept of CSPNet (Cross-Stage Partial Network), this backbone facilitates efficient information exchange between different stages of the neural network. This design choice improves the model's ability to capture complex hierarchical features, enhancing overall detection performance.

Key characteristics of the CSPDarknet53 backbone:

1. **Cross-Stage Partial Network (CSPNet):** The CSPNet architecture introduces the concept of cross-stage partial connections, which facilitates efficient information exchange between different stages or layers of the neural network. This design is particularly effective in capturing and utilizing contextual information across various scales and stages of the network.
2. **Backbone for Feature Extraction:** In the context of YOLOv8, the CSPDarknet53 backbone serves as the foundational neural network for feature extraction. Feature extraction involves transforming raw input images into a set of abstracted and hierarchical features that capture essential information for object detection.
3. **Hierarchical Feature Representation:** The CSPDarknet53 backbone focuses on creating a hierarchical representation of features. This means that the model can learn and extract features at different levels of abstraction, allowing it to understand both fine-grained details and more general contextual information within an image.

4. **Enhanced Contextual Understanding:** The CSP architecture enhances the model's ability to understand context by facilitating the flow of information between different stages. This is crucial for accurate object detection, especially in scenarios where objects may have complex structures or are partially occluded.
5. **Improved Detection Performance:** By incorporating the CSPDarknet53 backbone, YOLOv8 aims to improve the overall detection performance of the model. The backbone contributes to the model's capacity to handle diverse and challenging scenarios commonly encountered in real-world applications.

PANet Integration

PANet addresses challenges associated with object detection in cluttered or complex scenes. By aggregating information along different paths, PANet ensures that the model can effectively consider both local and global context when making predictions. This is particularly valuable for scenarios where objects may be partially obscured or where context is essential for accurate classification.

Key aspects of PANet integration:

1. **Contextual Information Aggregation:** The primary purpose of PANet is to aggregate contextual information from different paths or scales within the neural network. This is important for object detection, as it allows the model to consider both local and global context when making predictions. In situations where objects may be large or small relative to the overall image, PANet facilitates the incorporation of information from multiple scales to improve detection accuracy.
2. **Multi-Scale Feature Fusion:** PANet achieves contextual information aggregation through a process of multi-scale feature fusion. It enables the model to combine features from different levels of abstraction, ensuring that the network can effectively leverage information from both fine-grained details and more general contextual cues. This multi-scale approach is beneficial for accurately localizing and classifying objects in diverse settings.
3. **Adaptability to Object Size:** The ability to adapt to objects of varying sizes is a crucial aspect of effective object detection. PANet helps YOLOv8 address this challenge by allowing the model to dynamically adjust its focus and feature representation based on the scale of the objects present in the input image. This adaptability is particularly relevant in real-world scenarios where objects may exhibit considerable size variations.
4. **Improving Detection in Complex Scenes:** In complex scenes where multiple objects coexist and may overlap, PANet enhances the model's understanding of the spatial relationships between objects. By aggregating information across scales, the model becomes more robust to handling challenging scenarios, such as occlusions or densely populated scenes.
5. **Integration with YOLOv8 Architecture:** The PANet module is seamlessly integrated into the YOLOv8 architecture, complementing the existing components

like the CSPDarknet53 backbone. This integration reflects a holistic approach to improving the model's performance by combining advancements in feature extraction with effective contextual information aggregation.

YOLOv8-CSP

YOLOv8 also incorporates elements from YOLOv4-CSP, further optimizing its performance. YOLOv4-CSP introduced the CSPNet to the YOLOv4 architecture, enhancing the model's ability to handle complex scenes. YOLOv8 builds upon these improvements, refining the architecture for better real-time object detection.

Applications

The versatility of YOLOv8 makes it applicable across various domains. In autonomous vehicles, the model's ability to detect and track objects in real-time is crucial for ensuring safe navigation. Surveillance systems benefit from YOLOv8's capacity to identify and monitor objects in complex environments. Additionally, in robotics, where quick and accurate object detection is vital, YOLOv8's real-time capabilities shine.

Performance Metrics

Researchers have evaluated YOLOv8's performance using standard metrics such as mean average precision (mAP) and inference speed. The model consistently demonstrates competitive results, showcasing its efficiency in detecting objects across different scales while maintaining high precision.

Conclusion

In conclusion, YOLOv8 represents a significant advancement in real-time object detection. Through its incorporation of the CSPDarknet53 backbone, PANet, and lessons learned from previous versions, YOLOv8 strikes a compelling balance between speed and accuracy. Its multi-scale approach and adaptability to diverse applications make it a noteworthy contribution to the field of computer vision. As technology continues to evolve, YOLOv8 stands at the forefront of real-time object detection, with potential applications spanning autonomous systems, surveillance, and robotics. Researchers and practitioners alike are encouraged to explore and implement YOLOv8 in their projects, contributing to the ongoing progress in computer vision and artificial intelligence.