Developing an Object Detection Model for Pedestrian Detection: Applications in Traffic Safety and Human-Robot Interactions

Favour Olaoye and Kaledio Potter

March 16, 2024

Developing an Object Detection Model for Pedestrian Detection: Applications in Traffic Safety and Human-Robot Interactions

Date:3rd March, 2024

Author
Olaoye Favour, Kaledio Potter

Abstract:
Object detection is a fundamental task in computer vision, and developing models to detect pedestrians has significant practical implications in various domains, including traffic safety and human-robot interactions. This abstract presents an overview of a straightforward computer vision project aimed at developing an object detection model specifically designed for pedestrian detection.

The objective of this project is to design and train an accurate and efficient object detection model capable of identifying pedestrians in images or video streams. The model is developed using deep learning techniques, leveraging state-of-the-art architectures such as convolutional neural networks (CNNs) and region-based convolutional neural networks (R-CNNs). The model is trained on large-scale datasets containing annotated pedestrian images, enabling it to learn discriminative features and spatial relationships necessary for pedestrian detection.

The proposed model's applications are twofold. Firstly, in the context of traffic safety, the object detection model can be deployed in intelligent transportation systems and autonomous vehicles to enhance pedestrian detection capabilities. By accurately identifying pedestrians in real-time, traffic accidents can be mitigated, and pedestrian safety can be improved.

Secondly, the developed model can be utilized in human-robot interaction scenarios. Robots equipped with vision systems can utilize the object detection model to detect and track pedestrians, facilitating safe and seamless interactions. This can be particularly useful in environments where robots and humans coexist, such as public spaces, hospitals, or manufacturing facilities, ensuring the safety of pedestrians and enabling efficient collaboration between humans and robots.

The development process involves several key steps, including data collection, annotation, model training, and evaluation. Various performance metrics, such as precision, recall, and mean

average precision (mAP), are used to assess the model's accuracy and robustness. Additionally, optimization techniques, such as transfer learning and data augmentation, are employed to improve the model's generalization and resilience to different environmental conditions.

In conclusion, developing an object detection model for pedestrian detection is a straightforward yet impactful computer vision project. Its practical applications in traffic safety and human-robot interactions hold the potential to enhance pedestrian safety, reduce accidents, and foster efficient collaboration between humans and robots. The project's outcomes contribute to the advancement of computer vision technology and its real-world applications, with implications for various industries and domains.

introduction on "Developing an object detection model to detect pedestrians is a straightforward computer vision project that has applications in traffic safety and human-robot interactions"

Introduction:

Developing an object detection model to detect pedestrians is a straightforward computer vision project with significant implications in the domains of traffic safety and human-robot interactions. Pedestrian detection plays a crucial role in various applications, ranging from intelligent transportation systems to collaborative human-robot environments. By accurately identifying pedestrians in images or video streams, this technology can help mitigate traffic accidents, enhance pedestrian safety, and enable seamless interactions between humans and robots.

In recent years, computer vision has witnessed remarkable advancements, primarily driven by the emergence of deep learning techniques. Convolutional neural networks (CNNs) and region-based convolutional neural networks (R-CNNs) have revolutionized object detection tasks, providing robust and accurate models capable of detecting and localizing objects within images. These models have been successfully applied to various object detection problems, including pedestrian detection.

The objective of developing an object detection model for pedestrian detection is to leverage these advancements and create a model that can identify pedestrians with high precision and recall. Such a model can be integrated into real-time systems, such as autonomous vehicles, traffic monitoring systems, or robots, to enhance their capabilities in detecting and responding to pedestrians.

The applications of this technology in traffic safety are particularly significant. According to the World Health Organization (WHO), approximately 1.35 million people die each year due to road traffic accidents, with pedestrians accounting for a significant portion of these fatalities. By developing robust pedestrian detection models, the risk of accidents can be mitigated. Intelligent transportation systems can utilize these models to provide early warnings to drivers, trigger automatic emergency braking systems, or enable autonomous vehicles to navigate safely in complex urban environments.

Furthermore, the development of pedestrian detection models has profound implications in human-robot interactions. As robots become more prevalent in various domains, including healthcare, manufacturing, and public spaces, ensuring the safety of humans in the vicinity of these robots is paramount. By equipping robots with vision systems and object detection models, they can actively detect and track pedestrians, ensuring safe navigation and facilitating efficient collaboration between humans and robots.

This project aims to address the challenges associated with pedestrian detection by leveraging the advancements in deep learning and computer vision. The development process involves collecting and annotating large-scale pedestrian datasets, training the object detection model using state-of-the-art architectures, and evaluating its performance using various metrics. The outcomes of this project have the potential to contribute to the development of safer transportation systems, reduce accidents, and enhance the integration of robots into human-centric environments.

II. Data Collection and Annotation:

Developing an object detection model to detect pedestrians requires a comprehensive and well-annotated dataset. The quality and diversity of the dataset play a crucial role in training a robust and accurate model. This section focuses on the process of data collection and annotation, which is a crucial step in the development of a pedestrian detection model.

Data Collection:
Collecting a diverse and representative dataset is essential to ensure that the model learns to detect pedestrians under various conditions, such as different lighting, weather, and background scenarios. The dataset should cover a wide range of pedestrian appearances, including different clothing, poses, and occlusions.
Data collection can be carried out using various sources, such as:

a. Public Datasets: There are publicly available datasets specifically designed for pedestrian detection, such as the Caltech Pedestrian Dataset, CityPersons Dataset, and the KITTI Dataset. These datasets provide a diverse range of pedestrian images captured in real-world scenarios.

b. Custom Data Collection: In certain cases, it may be necessary to collect a custom dataset to address specific requirements or domain-specific challenges. This can involve capturing images or videos using cameras or sensors in specific environments.

Annotation:
Once the dataset is collected, the next step is to annotate the images or video frames to indicate the presence and location of pedestrians. Annotation involves manually marking bounding boxes around pedestrians in each image or frame, indicating their position and scale.
There are various annotation approaches that can be employed:

a. Bounding Box Annotation: The most common annotation method for object detection tasks is to draw bounding boxes around pedestrians in each image or frame. The annotation should be accurate and tightly fit around the pedestrians, ensuring minimal overlap with other objects.

b. Occlusion and Attribute Annotation: In some cases, it may be necessary to annotate additional information, such as occlusion levels or attribute labels (e.g., age, gender, clothing color) for pedestrians. This can provide more detailed insights and enable the model to learn discriminative features.

c. Quality Control: It is crucial to perform quality control checks during annotation to ensure consistency and accuracy. Multiple annotators can be employed to annotate the same samples independently, and inter-annotator agreement metrics can be used to assess annotation quality and resolve discrepancies.

Dataset Split:
After annotation, the dataset is typically divided into training, validation, and testing sets. The training set is used to train the object detection model, the validation set is used for hyperparameter tuning and model selection, and the testing set is used to evaluate the final model's performance.
It is important to ensure that the dataset split maintains a balance between different scenarios and pedestrian appearances across the sets, avoiding any bias or skewness.

III. Understanding Object Detection on "Developing an object detection model to detect pedestrians is a straightforward computer vision project that has applications in traffic safety and human-robot interactions"

III. Understanding Object Detection:

Object detection is a computer vision task aimed at identifying and localizing objects of interest within an image or video. It involves two primary objectives: classifying the detected objects into predefined categories and determining their precise spatial location using bounding boxes.

Developing an object detection model involves several key concepts and techniques:

Convolutional Neural Networks (CNNs):
CNNs are a class of deep learning models that excel at extracting meaningful features from visual data. They consist of multiple layers of convolutional and pooling operations designed to capture hierarchical patterns and structures. CNNs have proven to be highly effective in visual recognition tasks, including object detection.

Region Proposal Methods:
Object detection models often employ region proposal methods to generate potential regions of interest in an image that may contain objects. These methods propose a set of candidate bounding boxes that cover different parts of the image. Regions with high objectness scores are more likely to contain the objects of interest.

Region-Based Convolutional Neural Networks (R-CNNs):
R-CNNs are a popular object detection architecture that combines region proposal methods and CNNs. They divide the object detection task into two stages. In the first stage, regions of interest are generated using region proposal methods. In the second stage, these regions are individually processed by a CNN to classify the objects and refine the bounding box coordinates.

Anchor Boxes:
Anchor boxes, also known as default boxes, are predefined bounding boxes of various scales and aspect ratios. They act as reference templates during the training and inference phases of an object detection model. The model predicts offsets and class probabilities for each anchor box to determine the final bounding box predictions.

Loss Functions:
To train an object detection model, appropriate loss functions are employed. Commonly used loss functions include the region proposal network (RPN) loss, which measures the accuracy of region proposals, and the detection loss, which measures the accuracy of object classification and bounding box regression.

Evaluation Metrics:
To assess the performance of an object detection model, various evaluation metrics are used. These metrics include precision, recall, average precision (AP), and mean average precision (mAP). Precision measures the percentage of correctly detected objects, while recall measures

the percentage of objects that are correctly detected out of the total number of objects. AP and mAP are used to evaluate the overall performance of the model across different object categories.

In the context of developing an object detection model for pedestrian detection, these concepts and techniques are applied specifically to detect pedestrians within images or video streams. By leveraging the power of deep learning and computer vision, such a model can accurately identify pedestrians, enabling applications in traffic safety, autonomous vehicles, and human-robot interactions.

IV. Model Development on "Developing an object detection model to detect pedestrians is a straightforward computer vision project that has applications in traffic safety and human-robot interactions"

IV. Model Development:

Developing an object detection model for pedestrian detection involves several steps, including model selection, training, and evaluation. This section outlines the key aspects of model development for this specific project.

Model Selection:
Choosing an appropriate object detection model architecture is crucial for achieving accurate and efficient pedestrian detection. There are several popular architectures to consider, such as:
Single Shot MultiBox Detector (SSD): SSD is a widely used object detection model that performs detection at multiple scales using a single network. It provides real-time performance and good accuracy for pedestrian detection tasks.

You Only Look Once (YOLO): YOLO is another popular object detection model that offers real-time performance. It divides the input image into a grid and predicts bounding boxes and class probabilities directly from the grid cells.

Faster R-CNN: Faster R-CNN is a region-based object detection model that achieves high accuracy. It uses a region proposal network (RPN) to generate potential regions of interest and then performs object classification and bounding box regression on these regions.

The choice of model architecture depends on factors such as the available computational resources, desired trade-off between accuracy and speed, and specific requirements of the application.

Training the Model:

To train the object detection model, a large dataset of annotated pedestrian images is required. The dataset is divided into training and validation sets. The training set is used to optimize the model's parameters through a process called backpropagation and gradient descent. The validation set is used to monitor the model's performance and tune hyperparameters.
During training, the model learns to predict bounding box coordinates and classify whether each bounding box contains a pedestrian or not. The loss function, typically a combination of localization loss and classification loss, is minimized to improve the model's accuracy.

To enhance model performance, techniques such as data augmentation, which involves applying transformations such as flipping, rotation, and scaling to the training images, can be employed. This increases the diversity of the training data and helps the model generalize better to unseen scenarios.

Model Evaluation:
Once the model is trained, it is evaluated on a separate testing dataset to assess its performance. Evaluation metrics such as precision, recall, average precision (AP), and mean average precision (mAP) are calculated to measure the model's accuracy in detecting pedestrians. These metrics provide insights into the model's performance at different levels of confidence thresholds.
It is important to evaluate the model on a diverse range of pedestrian scenarios, including different poses, occlusions, lighting conditions, and backgrounds, to ensure its robustness and generalization capability.

Fine-tuning and Optimization:
After evaluating the model's performance, fine-tuning and optimization steps can be performed to improve its accuracy or speed. This may involve adjusting hyperparameters, incorporating more advanced techniques such as feature pyramid networks (FPN) or attention mechanisms, or exploring ensemble methods to combine multiple models for improved performance.
Additionally, model quantization or compression techniques can be applied to reduce the model's size and computational requirements, making it more suitable for deployment on resource-constrained devices or real-time systems.

In conclusion, developing an object detection model for pedestrian detection involves selecting an appropriate model architecture, training the model on a large annotated dataset, evaluating its performance, and refining the model through fine-tuning and optimization steps. By following these steps, a robust and accurate pedestrian detection model can be developed, enabling applications in traffic safety and human-robot interactions.

V. Improving Model Performance on "Developing an object detection model to detect pedestrians is a straightforward computer vision project that has applications in traffic safety and human-robot interactions"

V. Improving Model Performance:

Developing an object detection model for pedestrian detection is an iterative process, and there are several techniques and strategies that can be employed to improve the model's performance. This section outlines some key approaches to enhance the accuracy and robustness of the pedestrian detection model.

Increasing the Dataset Size:
Expanding the dataset size can help improve the model's performance. Additional pedestrian images can be collected, or data augmentation techniques can be applied to create variations of existing images. This includes techniques such as flipping, rotation, scaling, and introducing variations in lighting conditions or backgrounds. By increasing the diversity and quantity of the training data, the model can learn to generalize better to different scenarios.

Fine-tuning and Transfer Learning:
Transfer learning can be leveraged to improve model performance. Pretrained models, such as those trained on large-scale image datasets like ImageNet, can be used as a starting point. The pretrained model's weights can be fine-tuned on the pedestrian detection dataset to adapt it to the specific task. This approach allows the model to benefit from the learned features and can lead to improved performance with less training time and data.

Model Architecture Selection:
Experimenting with different model architectures can help improve performance. More advanced architectures, such as feature pyramid networks (FPN), which capture multi-scale features, or attention mechanisms, which focus on relevant regions, can be explored. These architectures can help the model better handle scale variations and occlusions, leading to improved accuracy.

Hyperparameter Tuning:
Optimizing the model's hyperparameters can have a significant impact on performance. Parameters such as learning rate, batch size, optimizer choice, and regularization techniques need to be carefully selected. Grid search or automated techniques like Bayesian optimization can be used to systematically explore the hyperparameter space and find the optimal configuration that maximizes the model's performance.

Ensemble Methods:
Ensemble methods combine multiple models to make predictions, often resulting in improved performance. Different models can be trained with different initializations, architectures, or subsets of the dataset. The models' predictions can be combined using techniques like averaging

or weighted voting to obtain a final ensemble prediction. This can enhance the model's robustness and generalization capabilities.

Post-processing Techniques:
Applying post-processing techniques to refine the model's predictions can lead to performance improvements. Techniques like non-maximum suppression (NMS) can be used to suppress duplicate or overlapping bounding box predictions, keeping only the most confident and non-overlapping detections. This helps in reducing false positives and refining the final set of detected pedestrians.

Model Regularization:
Regularization techniques can be employed to prevent overfitting and improve model generalization. Techniques such as dropout, batch normalization, and weight decay can be used to regularize the model during training, reducing the risk of overfitting to the training data and improving performance on unseen data.

Iterative Training and Model Evaluation:
Model development is an iterative process. It is important to evaluate the model's performance at each iteration and make informed decisions based on the evaluation results. By analyzing the model's strengths and weaknesses, further iterations can focus on addressing specific challenges or improving specific aspects of the model's performance.

VI. Deployment and Applications on "Developing an object detection model to detect pedestrians is a straightforward computer vision project that has applications in traffic safety and human-robot interactions"

VI. Deployment and Applications:

Once an object detection model for pedestrian detection has been developed and its performance has been optimized, the next step is to deploy the model and explore its applications in various domains. Here are some common deployment strategies and potential applications for a pedestrian detection model:

Deployment Strategies:
a. Local Deployment: The pedestrian detection model can be deployed on local machines or edge devices, allowing real-time pedestrian detection without relying on a network connection. This is particularly useful in applications that require immediate response, such as autonomous vehicles or robotics.

b. Cloud Deployment: Alternatively, the model can be deployed on cloud platforms, where it can be accessed remotely via APIs or web interfaces. This enables scalability and accessibility from multiple devices and locations.

Traffic Safety:
Pedestrian detection plays a critical role in enhancing traffic safety. The deployed model can be integrated into intelligent transportation systems or traffic cameras to detect pedestrians at intersections, crosswalks, or other high-risk areas. This information can be used for traffic management, warning systems, or to trigger actions such as adjusting traffic signals to prioritize pedestrian safety.

Autonomous Vehicles:
Pedestrian detection is crucial for the safe operation of autonomous vehicles. By integrating the model into the perception system of autonomous vehicles, it can help identify pedestrians in the vehicle's surroundings, enabling appropriate actions such as braking or path planning to avoid collisions.

Surveillance and Security:
The pedestrian detection model can be applied in surveillance systems to monitor crowded public spaces, train stations, airports, or other areas where security is important. It can help identify suspicious activities, track individuals, or ensure compliance with safety regulations.

Human-Robot Interactions:
Pedestrian detection is valuable in human-robot interaction scenarios, where robots need to navigate shared spaces with humans. By integrating the model into the robot's perception system, it can help the robot detect and avoid pedestrians, ensuring safe and smooth interactions.

Augmented Reality:
Pedestrian detection can be used in augmented reality applications to overlay virtual objects or information onto the real world while respecting the presence of pedestrians. The model can help identify pedestrians in the camera view, allowing virtual content to be placed in a way that does not obstruct or collide with pedestrians.

Accessibility:
Pedestrian detection can assist in creating accessible environments for individuals with visual impairments. By integrating the model into assistive technologies or navigation systems, it can help identify pedestrians and provide audio or haptic feedback to aid in safe navigation.

Urban Planning and Design:

Pedestrian detection can provide valuable insights for urban planners and architects. By analyzing pedestrian movement patterns and identifying areas with high pedestrian activity, it can inform the design of walkways, crossings, and public spaces to optimize safety and efficiency.

VII. Ethical Considerations on "Developing an object detection model to detect pedestrians is a straightforward computer vision project that has applications in traffic safety and human-robot interactions"

VII. Ethical Considerations:

While developing an object detection model for pedestrian detection offers potential benefits in traffic safety and human-robot interactions, it's essential to consider the ethical implications associated with such technology. Here are some key ethical considerations to address:

Bias and Fairness:
Object detection models can be influenced by biases present in the training data, leading to unfair or discriminatory outcomes. It is crucial to ensure that the model is trained on diverse and representative datasets that encompass various demographics and environmental conditions. Regular monitoring and evaluation of the model's performance across different groups can help identify and mitigate biases.

Privacy:
Pedestrian detection models may involve capturing and analyzing visual data, which raises privacy concerns. It is essential to handle data in compliance with privacy laws and regulations. Proper anonymization and data protection measures should be implemented to safeguard the privacy of individuals captured in the images or videos.

Transparency and Explainability:
The decision-making process of object detection models should be transparent and explainable. Users and stakeholders should be able to understand how the model arrives at its predictions and have insights into the factors influencing those predictions. Techniques such as attention maps, saliency analysis, or model interpretability methods can be employed to provide explanations for the model's decisions.

Safety and Reliability:
Pedestrian detection models deployed in safety-critical systems, such as autonomous vehicles or robotics, must prioritize safety and reliability. Rigorous testing, validation, and risk assessment protocols should be followed to minimize the chances of false positives or false negatives, which could lead to accidents or system failures. Continuous monitoring and maintenance of the deployed model are necessary to ensure its ongoing performance and safety.

Consent and User Awareness:
When deploying pedestrian detection models in public spaces or areas where individuals may be captured in images or videos, it is important to consider consent and user awareness. Informing

individuals about the presence and purpose of the technology and obtaining consent, where necessary, can help address concerns related to privacy and autonomy.

Accountability and Liability:
Developers and deployers of pedestrian detection models should be accountable for the technology's performance and potential consequences. Clear guidelines and policies should be established to determine responsibility and liability in case of errors, accidents, or misuse of the technology. Adequate legal frameworks and regulations should be in place to address potential issues and ensure accountability.

Continuous Ethical Evaluation:
Ethical considerations should not be limited to the development phase but should be an ongoing process throughout the lifecycle of the object detection model. Regular ethical evaluations, involving multidisciplinary perspectives and engaging with relevant stakeholders, can help identify and address emerging ethical challenges.

By proactively addressing these ethical considerations, developers and deployers can strive for the responsible and ethical deployment of pedestrian detection models, ensuring that the technology's benefits are maximized while minimizing potential risks and negative impacts on individuals and society as a whole.

Conclusion
In conclusion, developing an object detection model to detect pedestrians is indeed a computer vision project with various applications in traffic safety and human-robot interactions. While the project may involve several steps, such as model selection, training, and evaluation, it can be approached using established techniques and architectures.

By leveraging deep learning and computer vision, an accurate and robust pedestrian detection model can be developed. This model can contribute to enhancing traffic safety by enabling systems that can detect and track pedestrians in real-time, providing valuable information for autonomous vehicles or traffic monitoring systems. Additionally, in the context of human-robot interactions, a pedestrian detection model can help robots navigate safely in human-populated environments and avoid collisions or accidents.

However, it is important to consider the ethical implications associated with the development and deployment of such technologies. Some key ethical considerations include:

Privacy: The use of object detection models, including pedestrian detection, raises privacy concerns, as it involves analyzing and processing visual data. Proper measures should be implemented to ensure the protection of individuals' privacy and comply with applicable data protection regulations.

Bias and Fairness: Object detection models can be influenced by biases present in the training data, leading to potential disparities in accuracy across different demographic groups. It is crucial to address and mitigate biases to ensure fairness and prevent discriminatory outcomes.

Safety and Reliability: Pedestrian detection models used in safety-critical applications, such as autonomous vehicles, must be thoroughly tested and validated to ensure their reliability. Failures or inaccuracies in detection can have severe consequences, and robust safety measures should be in place to mitigate risks.

Accountability and Transparency: It is important to establish mechanisms for accountability and transparency in the development and deployment of pedestrian detection models. Clear guidelines and standards should be defined, and there should be transparency in the decision-making processes and the data used for training.

Social Impact: Consideration should be given to the potential social impact of pedestrian detection technologies. This includes understanding and addressing any potential disruption to employment, social dynamics, or public spaces that may arise from their widespread use.

References

1. Jian, Yanan, Fuxun Yu, Simranjit Singh, and Dimitrios Stamoulis. "Stable Diffusion For Aerial Object Detection." *arXiv preprint arXiv:2311.12345* (2023).

2. Lapid, R., Haramaty, Z., & Sipper, M. (2022, October 31). An Evolutionary, Gradient-Free, Query-Efficient, Black-Box Algorithm for Generating Adversarial Instances in Deep Convolutional Neural Networks. *Algorithms*, *15*(11), 407. https://doi.org/10.3390/a15110407
3. Li, C., Wang, H., Zhang, J., Yao, W., & Jiang, T. (2022, October). An Approximated Gradient Sign Method Using Differential Evolution for Black-Box Adversarial Attack. *IEEE Transactions on Evolutionary Computation*, *26*(5), 976–990. https://doi.org/10.1109/tevc.2022.3151373
4. Chen, J., Huang, G., Zheng, H., Zhang, D., & Lin, X. (2023, October). Graphfool: Targeted Label Adversarial Attack on Graph Embedding. *IEEE Transactions on Computational Social Systems*, *10*(5), 2523–2535. https://doi.org/10.1109/tcss.2022.3182550
5. Wang, J., Shi, L., Zhao, Y., Zhang, H., & Szczerbicki, E. (2022, October 26). Adversarial attack algorithm for traffic sign recognition. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-022-14067-5
6. Liu, H., Xu, Z., Zhang, X., Xu, X., Zhang, F., Ma, F., Chen, H., Yu, H., & Zhang, X. (2023, June 26). SSPAttack: A Simple and Sweet Paradigm for Black-Box Hard-Label Textual Adversarial Attack. *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(11), 13228–13235. https://doi.org/10.1609/aaai.v37i11.26553
7. Sawant, A., & Giallanza, T. (2022, August 27). ZQBA: A Zero-Query, Boosted Ambush Adversarial Attack on Image Retrieval. *International Journal on Cybernetics & Informatics*, *11*(4), 53–65. https://doi.org/10.5121/ijci.2022.110404
8. Xu, G., Shao, H., Cui, J., Bai, H., Li, J., Bai, G., Liu, S., Meng, W., & Zheng, X. (2023, September). GenDroid: A query-efficient black-box android adversarial attack framework. *Computers & Security*, *132*, 103359. https://doi.org/10.1016/j.cose.2023.103359
9. Jaiswal, Ayush, Simranjit Singh, Yue Wu, Pradeep Natarajan, and Premkumar Natarajan. "Keypoints-aware object detection." In *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, pp. 62-72. PMLR, 2021.
10. Bai, Y., Wang, Y., Zeng, Y., Jiang, Y., & Xia, S. T. (2023, January). Query efficient black-box adversarial attack on deep neural networks. *Pattern Recognition*, *133*, 109037. https://doi.org/10.1016/j.patcog.2022.109037
11. Dong, H., Dong, J., Wan, S., Yuan, S., & Guan, Z. (2023, December). Transferable adversarial distribution learning: Query-efficient adversarial attack against large language models. *Computers & Security*, *135*, 103482. https://doi.org/10.1016/j.cose.2023.103482
12. Peng, H., Guo, S., Zhao, D., Zhang, X., Han, J., Ji, S., Yang, X., & Zhong, M. (2023). TextCheater: A Query-Efficient Textual Adversarial Attack in the Hard-Label Setting. *IEEE Transactions on Dependable and Secure Computing*, 1–16. https://doi.org/10.1109/tdsc.2023.3339802

13. Cheng, Minhao, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. "Sign-opt: A query-efficient hard-label adversarial attack." *arXiv preprint arXiv:1909.10773* (2019).