Team Fight Tactics Assistant Tool

Occidental College, Computer Science Senior Comps

Jordan Jung

December 2023

1 Abstract

This project presents the development and implementation of an assistant tool for the strategy game Team Fight Tactics (TFT), utilizing pre-existing modules in computer vision and machine learning. The core objective is to enhance player decision-making by providing real-time insights into the game's unit pool, leveraging a custom-trained YOLO (You Only Look Once) v8 model for object detection and Tesseract OCR for text recognition. This tool differentiates itself from existing game assistants by eschewing direct API interactions in favor of an indirect, vision-based approach to data extraction. The project involved creating a hand-labeled dataset, training the YOLO model, and integrating these components into a user-friendly interface with an overlay system for in-game use. Evaluation of the tool's effectiveness was conducted through comparative gameplay analysis, with a focus on the tool's impact on strategic decisions and game outcomes. The results demonstrate a relatively significant improvement in gameplay performance when using the tool, highlighting its potential to alter strategic approaches in TFT. This work not only contributes to the field of AI in gaming but also opens discussions on the ethical implications of AI-assisted gameplay, particularly concerning fairness and player skill. The project's findings and methodologies offer valuable insights into the application of computer vision in interactive entertainment and lay the groundwork for future exploration in this domain.

2 Introduction

In this senior comps project, I explore the intersection of automation and computer vision applications in gaming, a burgeoning field with vast potential. Team Fight Tactics, a game involving strategy and quick decision-making, serves as the focal point for this exploration. Unlike existing tools that interact with the game's API, this project employs real-time computer vision to indirectly access game data, enhancing the player's strategic capabilities. This endeavor not only aims to improve the gaming experience but also contributes to the broader discourse on AI's role in recreational activities, particularly in balancing technological assistance with skill in gaming.

3 Technical Background

The Team Fight Tactics Assistant Tool is built upon three foundational technologies: Tesseract OCR, YOLO v8, and a user-friendly interface, complemented by a PyQt5-based overlay for real-time display.

3.1 Tesseract OCR

Tesseract OCR (Optical Character Recognition) is an open-source text recognition engine that specializes in extracting textual information from the game's interface. It is capable of recognizing text across various fonts and backgrounds, making it particularly useful for interpreting dynamic in-game elements. In this project, Tesseract OCR is employed to scan the TFT game screen, accurately identifying and extracting key textual data such as unit names, stats, and shop contents. This data is used for the tool's analysis and decision-making processes.

3.2 YOLO v8

YOLO (You Only Look Once) v8 is at the forefront of real-time object detection, utilizing a deep convolutional neural network. It contrasts with other object detection frameworks like SSD (Single Shot MultiBox Detector), which Liu et al. (2016) explored for its efficiency in detection tasks [4]. YOLO v8's integration of bounding box prediction and class probability calculation in a single step makes it ideal for the dynamic environment of Team Fight Tactics, offering both speed and accuracy [6].

3.3 User Interface

The user interface of the tool is designed for ease of use and accessibility. It allows players to interact with the tool's features, customize settings, and receive real-time insights. The interface is intuitive, ensuring that players can easily select units for tracking, adjust preferences, and interpret the data provided by the tool. This level of customization ensures that the tool can cater to a wide range of strategies and player preferences.

3.4 PyQt5 Overlay

The PyQt5 Overlay is a crucial component of the tool, providing a dynamic and interactive display overlayed on the game screen. Utilizing the PyQt5 library, the overlay features transparent, always-on-top windows that can display real-time data and insights directly within the game environment. This allows players to receive immediate feedback and analysis without disrupting their gameplay experience. The overlay is designed to be unobtrusive yet informative, enhancing the player's ability to make strategic decisions based on the tool's output.

3.5 Asynchronous Programming and Multi-Threading

The tool employs asynchronous programming and multi-threading to manage the concurrent operation of Tkinter and PyQt5, which are essential for the user interface and overlay system, respectively. Both frameworks require their own event loops to function correctly, necessitating an asynchronous approach. This multi-threaded design allows the tool to perform real-time data processing and user interaction simultaneously, enhancing responsiveness and user experience.

3.6 Integration and Real-Time Analysis

The integration of Tesseract OCR, YOLO v8, and the PyQt5 Overlay with the user interface allows for seamless real-time analysis of the game. As players engage in TFT matches, the tool continuously processes visual and textual data, providing up-to-date insights and recommendations. This integration is key to the tool's ability to offer statistics based on the current state of the game, helping players make informed decisions quickly and effectively.

In summary, the technical backbone of the Team Fight Tactics Assistant Tool combines advanced OCR and object detection technologies with a user-centric interface and an innovative overlay system. This combination enables the tool to offer real-time, data-driven insights, enhancing the strategic gameplay experience in TFT.

4 Literature Review

Prior research in computer vision in gaming has predominantly focused on enhancing player experience and automating gameplay. This project diverges from these traditional applications by integrating YOLO and OCR in Team Fight Tactics, a domain that has not been extensively explored. The literature review encompasses works on computer vision in gaming, real-time object detection algorithms, and OCR applications, providing a comprehensive background for the project.

The YOLO model, a cornerstone of this project, has been widely recognized for its efficiency and accuracy in real-time object detection. Its application in gaming, particularly in dynamic environments like RTS games, has been explored in various studies [1]. For instance, Redmon et al. introduced the YOLO model as a novel approach to object detection, emphasizing its speed and accuracy, which are crucial for real-time applications [6]. The model's ability to process images in real-time makes it an ideal choice for analyzing fast-paced game scenarios in TFT.

Tesseract OCR, another key component, has been utilized in various domains for text recognition. Its application extends beyond traditional document scanning to more interactive uses, such as in video games for extracting textual information from complex backgrounds. Smith's work on Tesseract OCR highlights its versatility and adaptability in different environments, making it a suitable tool for interpreting in-game text in TFT [8].

The integration of these technologies in gaming is still a relatively new venture. However, their successful application in other fields, such as automated data extraction and real-time analysis, provides a strong foundation for their use in Team Fight Tactics [2]. The project aims to leverage the strengths of YOLO and Tesseract OCR to create a unique gaming assistant that enhances strategic gameplay through data-driven insights.

Furthermore, research on an item recommendation system for League of Legends by Smit is of particular interest. Smit's use of an artificial neural network with various classifiers, coupled with the Local Interpretable Model-Agnostic Explanations (LIME) technique, presents a compelling approach [7]. LIME's ability to provide interpretable insights into classifier predictions could be instrumental in refining the TFT Assistant Tool, offering intelligent suggestions for more complex elements of gameplay such as unit itemization [9].

These studies collectively provide a rich foundation of knowledge and techniques that can be adapted and extended in the context of Team Fight Tactics, guiding the development of a tool that not only gathers statistics but also offers strategic insights based on deep learning and AI [5].

5 Methods

The methodology of the Team Fight Tactics Assistant Tool was multifaceted, encompassing the creation of a custom dataset, the application of machine learning models, and the development of a user interface and overlay system.

5.1 Dataset Creation and Annotation

The process of creating a custom dataset involved three primary steps:

1. **Data Collection:** Gathering image data from Team Fight Tactics, focusing on game states and scenarios that would be typical inputs for the tool. This included capturing various in-game elements such as the shop, board states, and unit interactions.

2. **Hand Labeling:** Using the Computer Vision Annotation Tool (CVAT), a significant portion of the collected data was hand-labeled to identify and categorize key game elements. This initial dataset served as the foundation for training the initial model.

3. **Semi-Supervised Learning:** The initial model was then employed to predict labels on a larger, unlabeled dataset. The predictions were manu-

ally reviewed and corrected, providing a feedback loop to refine and retrain the model, enhancing its accuracy and robustness.

5.2 Model Training and Implementation

The YOLO (You Only Look Once) v8 model was chosen for its efficiency in real-time object detection. Training and fine-tuning of the model were facilitated by ROBOFLOW, a platform that streamlined the process of uploading image and label data. The trained model was then implemented in a Python environment, specifically using Google Collaboratory for its accessible and powerful computing resources.

5.3 Overlay and Interface Development

- **Overlay System:** Developed using the PyQt5 library, the overlay system features full-sized, always-on-top, transparent windows. Dynamic text boxes within the overlay display real-time insights, such as unit scarcity, directly onto the game screen. The overlay interacts with the YOLO model to process and display data extracted from images of enemy boards.

- **User Interface:** The interface, built with the TKinter library, offers a robust and user-friendly platform for players to interact with the tool. It displays all champions in the game, allowing users to select specific units for highlighting during gameplay. The interface is integral to customizing the tool's functionality to suit individual player strategies.

5.4 Implementation of Asynchronous Programming

To effectively integrate the Tkinter interface and PyQt5 overlay, the tool utilizes asynchronous programming and multi-threading. This approach ensures that both frameworks can run their event loops independently without blocking each other, allowing for seamless real-time updates and interactions. The asynchronous nature of the tool is critical in maintaining high performance and responsiveness during gameplay.

5.5 Alternative Approaches and Tool Optimization

While alternative methods such as pre-trained models or simpler detection algorithms were initially considered, they were ultimately not pursued due to their limitations in specificity and real-time processing capabilities. The project also emphasized streamlining the tool into easily installable modules, enhancing user accessibility and experience. This approach ensured that the tool remained adaptable and efficient, suitable for the dynamic environment of Team Fight Tactics.

In summary, the methods employed in this project combined advanced machine learning techniques with practical application development, resulting in a tool that not only gathers and analyzes data but also presents it in an accessible and actionable manner for TFT players.

6 Evaluation

The tool's effectiveness was evaluated based on its impact on gameplay decisions and the number of games won. Data was collected over a series of 10 games, documenting player performance and subjective opinions on the tool's usefulness. This metric reflects the tool's impact on enhancing gameplay strategy. Given that the player's skill level is a factor in the tools effectiveness, these metrics and resulting analyses can vary from player-to-player. For the most part, the information is generalized enough such that most users will benefit from it regardless, however, more experienced players can infer a lot more with the same data. Other metrics like decision-making speed were considered but not used due to quantification challenges.

7 Results and Discussion

The results showed an increase in Top 4 placements when using the tool compared to the control set. This suggests that the tool's insights into unit availability significantly influenced player strategy. More specifically, it affirms evidence that there is a rudimentary correlation between unit scarcity and player success. Where the aggregation of this data isn't prohibited, as the player can make these same inferences themselves, its clear that its ability to individuate particular units helps more than general observations about the content of enemy boards.

User feedback indicated the usefulness of the statistical data, though its full potential was realized only with effective data interpretation. The project highlighted the need for a more robust dataset to reduce false positives and the importance of tuning hyperparameters for model accuracy. In addition to improving the current dataset, I'd like to explore different ways in optimizing or automating the dataset creation process. Given that the champions in the game are rotated every 3 months, the model quickly becomes obsolete until retrained. Thus, streamlining the process would result in less work and sooner usage when a rotation occurs.

8 Ethical Considerations

Personal Ethics

The development of the Team Fight Tactics Assistant Tool was guided by a commitment to ethical principles, particularly in maintaining the integrity of competitive eSports. As a developer, my aim was to create a tool that enhances learning and strategic understanding for players, rather than providing an unfair advantage. This aligns with my personal belief in the importance of fair play and the educational value of gaming. The tool is intended for private use and educational purposes, reflecting a responsible approach to the application of AI in gaming environments.

Professional Ethics

Crawford and Calo (2016) highlight a blind spot in AI research, particularly concerning ethical implications [3]. In the realm of professional eSports, the tool parallels the role of a coach, offering strategic insights based on game data. However, it's crucial to ensure that its use remains within ethical boundaries, especially in professional settings. The tool is designed to analyze game states and suggest strategies, but it should not be used to gain unfair advantages during live matches. Upholding the ethical standards of professional eSports, the tool is intended as a supplement to player skill and strategy, not a replacement. Its development and potential use in professional scenarios were carefully considered to avoid any form of misuse that could undermine the spirit of fair competition.

Social/Political Ethics

The social and political implications, particularly in the context of AI's role in gaming and entertainment, are diverse. The tool's development raises questions about the democratization of advanced gaming strategies through technology. While the tool aims to provide strategic insights, it's crucial to consider its accessibility and the potential to create disparities in the gaming community. If the tool becomes widely used, it could lead to a shift in how games are played, potentially favoring those with access to such technologies.

Moreover, the reliance on AI and machine learning in gaming also brings into focus issues of data privacy and the ethical use of player-generated data. Ensuring that the tool respects player privacy and adheres to data protection regulations is paramount. The project also touches upon the broader debate of AI assistance in personal entertainment – where to draw the line between player skill and technological aid.

In developing this tool, care was taken to align with ethical standards that respect the integrity of the game and its community. Future developments will continue to consider these social and political aspects, aiming to contribute positively to the gaming culture without compromising the core values of fair play and equal opportunity for all players.

9 Conclusion

This project successfully demonstrates the innovative application of computer vision and machine learning in enhancing strategic gameplay in Team Fight Tactics (TFT). By integrating technologies such as Tesseract OCR and YOLO v8, along with a user-friendly interface and dynamic PyQt5 overlay, the tool has shown a significant potential to transform how players interact with and strategize in TFT. The increase in games won by players using the tool is a testament to its effectiveness in providing real-time, data-driven insights.

As Yannakakis and Togelius (2018) discuss in "Artificial Intelligence and Games," the integration of AI in gaming is reshaping game culture and strategy

[10]. This project's implementation of computer vision and machine learning in TFT is a testament to this evolution, offering new strategic dimensions to players. It showcases the potential of these technologies to not only enhance player experience but also to contribute to the development of more intelligent and responsive gaming tools. This project serves as a stepping stone towards a future where AI can work in tandem with players, offering a blend of strategic depth and accessibility.

Looking forward, there are several areas for further research and development:

- **Refining Model Accuracy:** Continuous improvement of the YOLO model and OCR accuracy to ensure more precise and reliable game analysis.
- **Dataset Expansion:** Expanding the dataset to include a wider range of game scenarios and player strategies, enhancing the tool's applicability to different skill levels.
- **User Experience:** Further development of the user interface and overlay system to provide more personalized and intuitive interactions for players.
- **Ethical Considerations:** Ongoing assessment of the tool's impact on gaming fairness and player privacy, ensuring responsible use of AI in gaming.
- **Adaptability to Game Updates:** Ensuring the tool remains effective and relevant with the frequent updates and changes in TFT and similar games.

In conclusion, the Team Fight Tactics Assistant Tool not only achieves its immediate objective of enhancing gameplay strategy but also contributes valuable insights into the integration of AI in gaming. It paves the way for future innovations in this field, promising a new era of AI-assisted gaming that is both challenging and accessible to a broad spectrum of players.

10 Appendices

10.1 Appendix A: Replication Instructions

To replicate and use the Team Fight Tactics Assistant Tool, follow these steps:

Step 1: Environment Setup

- Ensure Python 3.x is installed on your system.
- Install necessary libraries: PyQt5, Tesseract OCR, TensorFlow, OpenCV, and other dependencies listed in the 'requirements.txt' file.

Step 2: Clone the Repository

- Clone the project repository from GitHub using the command: 'git clone https://github.com/jjung2-oxy/TFTbot2.0'.
- Navigate to the cloned directory.

Step 3: Running the Tool

- Execute the main script in terminal to start the tool: 'python main.py'.
- Adjust preferences using the user interface.

Note: Detailed instructions and troubleshooting tips are available in the 'README.md' file in the repository.

10.2 Appendix B: Code Architecture Overview

The Team Fight Tactics Assistant Tool's codebase is structured as follows:

Main Components:

- **main.py:** The entry point of the application. It initializes the tool and integrates different modules.
- **threaded_main.py:** This file is invoked by the main file, specifically ran on a separate thread from the TKinter Window and Overlay.
- **interface.py:** Manages the user interface, built with TKinter, for user interaction and settings configuration.
- **overlayNEW.py:** Handles the PyQt5 overlay system for displaying real-time insights on the game screen.

Machine Learning Modules:

- **image_inference.py:** Contains the implementation and configuration of the YOLO v8 model for object detection.
- **OCR.py:** Integrates Tesseract OCR for text extraction and processing.

Utility Scripts:

- **champs_list.py:** Variables containing the names and cost of all Set 10 units.
- **screen_coords.py:** Includes fixed screen coordinates for the program to refer to.

Asynchronous Architecture:

- The application is structured to run Tkinter and PyQt5 on separate threads, enabling them to operate their event loops concurrently.

- Asynchronous programming techniques are employed to manage these threads, ensuring smooth and responsive operation of both the user interface and the overlay system.
- This multi-threaded approach is integral to the tool's ability to process and display data in real-time, providing a dynamic and interactive user experience.

Note: Each module is well-documented with comments explaining the functionality and interactions. For further development or customization, refer to the inline comments and documentation within each script.

10.3 Appendix C: Additional Resources

- **Project Repository:** Access the full source code and documentation on GitHub at https://github.com/jjung2-oxy/TFTbot2.0.

References

[1] Christopher Amato and Guy Shani. High-level reinforcement learning in strategy games. In *AAMAS*, volume 10, pages 75–82, 2010.

[2] Paul Bertens, Anna Guitart, Pei Pei Chen, and Africa Perianez. A machine-learning item recommendation system for video games. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–4. IEEE, 2018.

[3] Kate Crawford and Ryan Calo. There is a blind spot in ai research. *Nature*, 538(7625):311–313, 2016.

[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[7] Robin Smit. *A machine learning approach for recommending items in League of Legends*. PhD thesis, Bachelors thesis, 2019.

[8] Ray Smith. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

[9] Vatsal. Recommendation systems explained, May 2022.

[10] Georgios N Yannakakis and Julian Togelius. *Artificial Intelligence and Games*. Springer, 2018.