

 <b>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</b>	Tipo de Prova Trabalho Prático – Av. em período letivo	Ano letivo 2024/2025	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Inteligência Artificial	Duração	

Observações

- Versão inicial (1.0)

### A. Destinatários

Este trabalho prático destina-se a todos os estudantes inscritos na Unidade Curricular de Inteligência Artificial, da Licenciatura em Engenharia Informática, que pretendam fazer a componente prática da sua avaliação durante o período letivo.

O trabalho prático tem um peso de 50% na classificação final da UC e tem como requisito mínimo uma classificação de 9.5 valores

### B. Objetivos

Este projeto funcionará como um elemento integrador dos conhecimentos adquiridos na UC de Inteligência Artificial, com um foco especial na área de Machine Learning e Visão por Computador. Nomeadamente, serão trabalhadas competências fundamentais na área de IA, incluindo:

- A modelação do conhecimento existente no domínio de um problema e no seu espaço de solução, com vista à sua utilização computacional;
- A geração, otimização e avaliação de soluções válidas para um problema complexo, utilizando uma perspetiva iterativa de desenvolvimento e validação;
- A análise e comparação crítica de diferentes abordagens, com vista à seleção da mais adequada à resolução do problema;
- A melhoria iterativa e incremental de uma abordagem para a resolução de um problema com base em resultados passados;
- A criação de datasets para problemas específicos de Machine Learning;
- A utilização de algoritmos de Machine Learning para treino de modelos, e a utilização em produção;

Especificamente, o principal objetivo deste trabalho prático é o desenvolvimento de uma aplicação que, através de modelos de visão computacional baseados na arquitetura YOLO (You Only Look Once), e mais especificamente modelos de deteção de objetos, seja capaz de automatizar alguma tarefa num computador ou aplicação. O caso de uso sugerido é que os estudantes escolham um jogo que conheçam, e automatizem uma ou mais tarefas nesse jogo. Idealmente, cada elemento do grupo deveria focar-se na implementação de uma tarefa diferente, ficando responsável pelo desenvolvimento de um modelo específico para essa tarefa, não obstante o conjunto de imagens a recolher e a aplicação a desenvolver serem comuns.

Alguns exemplos incluem:

- Assistência de mira (aim-bots) - Utilizados em FPS (First-Person Shooters) como Counter-Strike, Call of Duty ou Valorant. O modelo deteta inimigos no ecrã, e ajusta automaticamente a mira do jogador e pode eventualmente disparar.
- Automação de recolha de recursos (farming bots) - Aplicável em jogos como FarmVille, Forge of Empires, World of Warcraft ou Genshin Impact. O modelo pode identificar elementos como colheitas maduras, minérios ou baús e automatizar a ação de recolha.

 <b>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</b>	Tipo de Prova Trabalho Prático – Av. em período letivo	Ano letivo <b>2024/2025</b>	Data
	Curso <b>Licenciatura em Engenharia Informática</b>	Hora	
	Unidade Curricular Inteligência Artificial	Duração	

- Deteção e resolução automática de quebra-cabeças. Aplicável em jogos como Candy Crush, Bejeweled ou Tetris. O modelo pode reconhecer padrões no jogo e sugerir ou realizar a melhor jogada, nomeadamente integrando com técnicas de otimização ou recuperando informação de bases de conhecimento.
- Automação de minijogos ou eventos temporários. Em jogos como Animal Crossing, Runescape ou GTA Online, é possível detetar eventos específicos no jogo (como um peixe a morder o isco), acionando uma resposta automática.
- Reconhecimento de cartas ou objetos em jogos de estratégia. Aplicável em jogos como Hearthstone, Magic: The Gathering Arena ou Yu-Gi-Oh! Duel Links, em que o modelo pode identificar cartas na mão do jogador ou do oponente e sugerir jogadas otimizadas, em conjunto com técnicas de otimização.
- Automação de movimentação baseada em deteção de objetos. Útil em jogos de exploração como The Legend of Zelda: Breath of the Wild, Red Dead Redemption 2 ou Dark Souls, em que o modelo pode detetar caminhos, inimigos, obstáculos ou outros objetos para guiar um personagem de forma autónoma (e eventualmente guiar a mira). Por exemplo, para capturar/caçar animais específicos.

### C. Disclaimer

Este trabalho prático tem fins estritamente académicos e destina-se exclusivamente ao estudo e à aplicação de modelos de visão computacional, como o YOLO, no contexto da automação de tarefas em jogos ou outras aplicações eletrónicas.

Os estudantes devem estar cientes de que muitos jogos possuem sistemas anti-cheat e termos de serviço que proíbem qualquer forma de automação, modificação ou interferência no funcionamento do jogo. Empresas como a AnyBrain (<https://anybrain.gg/>), da qual o docente é co-fundador, e outras desenvolvedoras de sistemas anti-cheat monitoram e penalizam práticas que violem essas regras.

Antes de escolher um jogo para este trabalho, cada estudante deve verificar os termos de uso do jogo e garantir que a atividade proposta não infringe nenhuma política. A responsabilidade pelo cumprimento dessas diretrizes é inteiramente do estudante. Qualquer uso indevido deste conhecimento fora do contexto académico pode ter consequências legais e éticas. Aconselha-se ainda, especialmente no caso de jogos FPS, que o trabalho a desenvolver seja apenas utilizado no contexto de partidas casuais (não competitivas) e/ou no contexto de servidores privados ou com bots, que tipicamente não são monitorizados por sistemas de anti-cheat.

### D. Enunciado

O desenvolvimento de um projeto de visão computacional passa por um conjunto de fases típicas, que devem também ser seguidas na implementação deste trabalho. É ainda frequente que este processo tenha várias iterações. Ou seja, que em certos pontos do processo se regresse a fases anteriores ou mesmo ao início, de forma a melhorar algum dos aspetos já desenvolvidos. As próximas secções descrevem o trabalho que deve ser desenvolvido, de forma a dar resposta a este enunciado e aos objetivos propostos. Estas secções definem também a estrutura sugerida para o relatório a elaborar.

#### 1. Definição do Problema

O trabalho deve principiar por identificar um jogo ou aplicação eletrónica, bem como de uma ou mais tarefas que se pretendam automatizar. Cada grupo deve identificar e descrever claramente o jogo/aplicação, bem como a

<b>P.PORTO</b>  <b>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</b>	Tipo de Prova Trabalho Prático – Av. em período letivo	Ano letivo 2024/2025	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Inteligência Artificial	Duração	

tarefa ou tarefas a implementar, fundamentando a sua relevância no contexto do jogo/aplicação. Isto é, idealmente, devem ser automatizadas tarefas que tenham elevado valor e impacto na lógica de jogo ou de utilização da aplicação. Nesta fase, deve ainda ser caracterizada a tarefa ou tarefas a desenvolver, nomeadamente na sua implicação na fase de etiquetagem dos dados, bem como antever como o modelo a desenvolver poderá ser integrado no jogo/aplicação, para automatizar a tarefa.

## 2. Recolha de imagens

A recolha de um número suficiente de imagens para a tarefa a resolver é um passo fundamental no desenvolvimento de modelos de visão por computador. Estas imagens devem conter um conjunto diversificado e equilibrado de representações dos objetos ou elementos a detetar pelo modelo. Neste contexto, “diversificado” e “equilibrado” significa ter não apenas todos os objetos igualmente representados, mas ter variações na sua representação que incluem, entre outros e quando aplicável: diferentes ângulos, diferentes níveis de zoom, diferentes backgrounds, diferentes cores/brilho, etc. Um conjunto de imagens pequeno e/ou pouco diverso pode levar o modelo a ajustar-se demasiado bem a esse conjunto de dados (*overfit*), e tendo uma performance significativamente pior em novos conjuntos de dados. Isto é, um modelo com pouca capacidade de generalização.

Considerando as características do modelo a utilizar, recomenda-se que as imagens a recolher tenham uma resolução de 640x640. Esta resolução não é obrigatória, uma vez que as imagens podem ser redimensionadas antes de serem passadas ao modelo. Isto pode, no entanto, levar a que ocorram distorções na imagem, que venham a prejudicar a visibilidade dos elementos e a consequente qualidade do modelo.

## 3. Etiquetagem

A tarefa que será implementada no contexto deste trabalho prático, por ser uma tarefa de aprendizagem supervisionada, requer a existência de um dataset etiquetado. Mais especificamente, por se tratar de uma tarefa de deteção de objetos, é necessário, para cada objeto em cada imagem:

- Identificar os seus limites (bounding box)
- Identificar a sua classe, i.e., que tipo de objeto é

Isto pode ser feito em diferentes aplicações disponíveis gratuitamente, e tem como objetivo passar informação ao modelo sobre as características e localização de cada objeto a detetar na imagem, de forma que possa aprender os padrões relevantes. Não sendo de utilização obrigatória, no contexto deste trabalho prático sugere-se a utilização da aplicação roboflow, disponível em <https://roboflow.com/>. Esta plataforma permite ainda a obtenção do dataset a partir da aplicação python que será desenvolvida posteriormente, para treino do modelo.

No processo de etiquetagem devem ser seguidos os seguintes passos:

- Registo na plataforma roboflow e criação de um workspace
- Criação de um projeto do tipo Object Detection
- Upload das imagens previamente recolhidas
- Etiquetagem de todos os objetos nas imagens (Annotate)
- Criação de um dataset, com uma divisão entre train, test e split

A Fig. 1 ilustra um exemplo do resultado de etiquetagem de uma imagem, em que se pretende desenvolver um modelo que faça a recolha automática de recursos quando estes estão prontos.

<b>P.PORTO</b> <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático – Av. em período letivo	Ano letivo 2024/2025	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Inteligência Artificial	Duração	



Figura 1: Exemplo de etiquetagem, tendo 4 objetos de um tipo (tools, a roxo) e 1 objeto de outro tipo (coins, a vermelho).

#### 4. Data Augmentation

Esta fase, de cariz opcional, tem como principal objetivo aumentar a diversidade do dataset, de forma a obter um modelo mais robusto e generalizável. Ou seja, a partir de um conjunto reduzido de imagens, será produzido de forma automática um conjunto alargado de imagens, aplicando transformações nas imagens originais.

Dependendo do dataset e do problema a resolver, um processo de data augmentation pode incluir transformações como rotações, espelhos, alterações de brilho ou cor, efeitos de desfocagem, etc. Todas estas transformações têm como objetivo ou aumentar a diversidade dos dados, ou obrigar o modelo a aprender em condições mais difíceis, levando-o a aprender melhor os padrões existentes nas imagens.

As tarefas de data augmentation podem ser implementadas pelo grupo, ou em alternativa podem ser utilizadas as tarefas fornecidas pelo roboflow. No entanto, apenas devem ser implementadas tarefas de data augmentation que, em função das características do problema, façam sentido.

#### 5. Treino do Modelo

O processo de treino do modelo é a fase central do seu desenvolvimento, e toma como input o conjunto de dados previamente etiquetado. Para facilitar o trabalho a desenvolver nesta fase, é fornecido juntamente com este enunciado um repositório gitlab que tem um exemplo funcional do treino de um modelo YOLOv8. O repositório

 <b>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</b>	Tipo de Prova Trabalho Prático – Av. em período letivo	Ano letivo 2024/2025	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Inteligência Artificial	Duração	

está disponível em [https://gitlab.estq.ipp.pt/dcarneiro/ia\\_24\\_25](https://gitlab.estq.ipp.pt/dcarneiro/ia_24_25) (requer utilização da VPN quando fora da ESTG).

O caso de uso que serve de exemplo, é a recolha de recursos no popular jogo Forge of Empires. Para este efeito, foi criado um pequeno dataset, composto por 19 imagens, fornecido em anexo a este enunciado. Na aula prática será feito um tutorial da ferramenta roboflow com base nestas imagens, e será treinado um modelo, a título de exemplo, seguindo o projeto fornecido.

Na utilização do projeto fornecido (notebook python), devem ser tidas em consideração as instruções fornecidas no README do repositório.

## 6. Avaliação do Modelo

Nesta fase, o grupo deve reportar e analisar os resultados obtidos no treino do modelo. Isto deve ser feito com base em métricas objetivas típicas de projetos de deteção de objetos, que devem ser escolhidas pelo grupo, de forma fundamentada. Podem/devem ser ainda utilizados os elementos que a biblioteca ultralytics cria durante e após o treino do modelo, para avaliação da sua qualidade.

Com base nesta análise, o grupo deve decidir se o modelo é suficientemente bom para ser colocado em produção ou não. Em caso negativo, o grupo deve repetir partes ou todo o processo, na tentativa de melhorar o modelo.

Em qualquer caso, deve ser reportado no relatório todo o processo, incluindo as falhas e repetições, e não apenas a última instância.

## 7. Implementação em Produção

A última fase do projeto passa pelo desenvolvimento de uma aplicação que utilize o modelo desenvolvido em tempo real. Esta aplicação deverá, assim, ter como objetivo implementar a(s) tarefa(s) identificada(s) na fase 1. Para o fazer, a aplicação deverá ser capaz de capturar o conteúdo do ecrã e passá-lo como input ao modelo, de forma que este possa devolver uma deteção, para que a aplicação faça algo com esse resultado.

Para esse efeito, sugere-se a utilização da biblioteca mss, sendo que é possível capturar a imagem e passá-la diretamente ao modelo, sem a necessidade de a gravar como ficheiro.

É natural que, dependendo da tarefa definida, tenha ainda de ser controlado o rato e/ou o teclado para executar alguma ação. Isto pode ser feito com bibliotecas como a pyautogui.

A parte final do notebook fornecido exemplifica isto. Especificamente, o script automatiza a recolha de recursos no jogo Forge of Empires, com base no modelo treinado previamente. Para o efeito, o script monitoriza o ecrã a intervalos regulares e, sempre que é detetado um ícone pelo modelo, o rato é movido até ao centro da bounding box e o recurso recolhido através de um clique.

A Figura 2 mostra o resultado da utilização do modelo para prever num print do jogo. No caso, são detetados alguns objetos corretamente, mas existem também dois falsos positivos, embora com um baixo grau de confiança pelo modelo. Neste caso, foi utilizado um valor mínimo de confiança de 0.1. Claramente, este poderia ser ajustado para um valor superior, para minimizar o número de falsos negativos, sem aparentemente afetar o número de verdadeiros positivos.

P.PORTO  ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	Tipo de Prova Trabalho Prático – Av. em período letivo	Ano letivo 2024/2025	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Inteligência Artificial	Duração	



Figura 2: Exemplo de utilização do modelo em produção. A imagem mostra várias deteções e o seu nível de confiança. Existem dois falsos positivos, que poderiam ser removidos utilizando-se um nível de confiança maior (foi usado o valor 0.1).

## E. Conteúdos de Suporte

Tal como já descrito anteriormente, e de forma a facilitar a preparação de um projeto de deteção de objetos usando técnicas de visão computacional, é fornecido juntamente com este enunciado um repositório gitlab que tem um exemplo funcional do treino de um modelo YOLOv8, disponibilizado na forma de um notebook python.

O repositório está disponível em [https://gitlab.estg.ipp.pt/dcarneiro/ia\\_24\\_25](https://gitlab.estg.ipp.pt/dcarneiro/ia_24_25) (requer utilização da VPN quando fora da ESTG). O repositório contém ainda um ficheiro README, com as instruções de instalação e configuração. Não sendo de utilização obrigatória, sugere-se que os estudantes tomem este repositório como base para o trabalho a desenvolver.

## F. Requisitos Mínimos e Critérios de Avaliação

Esta secção descreve os requisitos mínimos/obrigatórios bem como os critérios de avaliação. Nesse sentido, o relatório a desenvolver e a entregar juntamente com o projeto, deve indicar claramente como o grupo cumpriu (ou não) cada um destes requisitos.

Os seguintes pontos são requisitos obrigatórios para a obtenção de nota mínima nesta componente da avaliação:

- Treinar pelo menos 2 modelos diferentes, de forma a compará-los criticamente. Os diferentes modelos podem ser obtidos através da utilização de diferentes conjuntos de imagens, através da utilização de

<b>P.PORTO</b>  <b>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</b>	Tipo de Prova Trabalho Prático – Av. em período letivo	Ano letivo 2024/2025	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Inteligência Artificial	Duração	

diferentes técnicas de data augmentation, ou através da utilização de diferentes hiper-parâmetros. Para este último ponto, sugere-se a consulta da documentação da biblioteca ultralytics (<https://docs.ultralytics.com/guides/hyperparameter-tuning/>), que usa Algoritmos Genéticos (a abordar futuramente no contexto desta UC) para pesquisar o melhor conjunto de hiper-parâmetros.

- Comparar de forma crítica e fundamentada os modelos treinados, para ser capaz de escolher de forma objetiva qual o melhor modelo para colocar em produção. Esta comparação deve ser feita com base em métricas objetivas, visualizações e outros elementos relevantes para avaliação da performance preditiva dos modelos.
- Colocar pelo menos um modelo treinado em produção. Isto implica o desenvolvimento de uma aplicação que implemente pelo menos uma das tarefas identificadas pelo grupo, utilizando o modelo em tempo real para a sua automatização.

A nota do trabalho prático (NTP) será dada pelos seguintes elementos principais, cujas componentes são descritas de seguida:

$$NTP = 0.2A + 0.1B + 0.3C + 0.1D + 0.3E$$

Em que:

- NTP – Nota do Trabalho Prático.
- A – Recolha de imagens e etiquetagem. Serão avaliadas a qualidade, quantidade e diversidade do dataset recolhido, bem como da sua adequação ao problema a resolver. Aspetos como a introdução de viés nos dados ou o seu desbalanceamento serão avaliados negativamente.
- B – Data augmentation. Será avaliada a relevância e aplicabilidade das técnicas de data augmentation aplicadas e/ou implementadas, bem como o seu contributo para a melhoria da qualidade dos modelos.
- C – Treino e avaliação do modelo. Será avaliada a qualidade e validade das metodologias utilizadas no treino dos modelos, a extensão da pesquisa de hiper-parâmetros, bem como a qualidade da avaliação dos resultados.
- D – Implementação em produção. Neste ponto avalia-se a qualidade e utilidade da aplicação desenvolvida para resolver a(s) tarefa(s) propostas pelo grupo, bem como o número e complexidade das tarefas resolvidas.
- E – Relatório + apresentação. Este ponto avalia a qualidade geral do relatório desenvolvido, bem como a prestação individual durante a apresentação e defesa.

Não obstante tratar-se de um trabalho de grupo, a nota é individual.

## G. Relatório

Juntamente com o projeto, deverá ser entregue um relatório escrito que detalhe os aspetos mais relevantes do trabalho desenvolvido. O relatório deverá ter, no máximo, 20 páginas, e sugere-se que seja estruturado de acordo com as fases descritas na secção D deste enunciado. Para além das secções recomendadas, o relatório deve ainda ter, pelo menos, uma secção introdutória e outra de conclusões, bem como qualquer outra secção que o grupo ache pertinente.

<b>P.PORTO</b>  <b>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</b>	Tipo de Prova Trabalho Prático – Av. em período letivo	Ano letivo 2024/2025	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Inteligência Artificial	Duração	

## H. Elementos a entregar

Cada grupo deverá entregar um ficheiro .zip cujo nome segue a convenção IA\_XXX.zip, em que XXX representa o número do grupo. O .zip deverá ter o seguinte conteúdo:

- **Relatório\_IA\_XXX.pdf** – relatório em pdf, detalhando todos o trabalho desenvolvido, com um máximo de 20 páginas
- **Projeto\_IA\_XXX** – pasta contendo o projeto desenvolvido e que contém todo o código relevante, quer da aplicação final, quer scripts de treino e avaliação de modelos, etc. O projeto deve ainda conter um README com as instruções para a execução dos scripts/aplicações
- **Dataset\_IA\_XXX** – pasta contendo todas as imagens recolhidas com a sua divisão para treino, teste e validação, bem como os ficheiros de etiquetagem correspondentes
- **TEAM\_IA\_XXX.txt** – ficheiro de texto contendo a identificação (nome e número) dos elementos que constituem o grupo

## I. Realização

Este trabalho é realizado em grupos de 3 elementos, salvo situações particulares a validar previamente com o docente da UC.

Os grupos devem ser escolhidos via Moodle, até à data limite de 25/04/2025.

O trabalho desenvolvido deve ser entregue na página da UC no Moodle, até às 23:59 do dia 27 de maio de 2025.

Apenas um elemento de cada grupo deverá submeter o trabalho em nome do grupo.

A apresentação e defesa do trabalho, de carácter obrigatório e em que todo o grupo terá de estar presente simultaneamente, decorrerá nas aulas do dia 2 de junho de 2025, em horários a agendar previamente com cada grupo.