



Instituto Superior de Engenharia

Politécnico de Coimbra

DEPARTAMENTO DE INFORMÁTICA E SISTEMAS

Redes Neurais Conhecimento e Raciocínio

Relatório de Licenciatura

Autores

Ana Rita Conceição Pessoa – 2023112690

João Francisco de Matos Claro – 2017010293



INSTITUTO POLITÉCNICO DE
COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, abril de 2025

1 ÍNDICE

1.1 Índice de texto

1	Índice.....	1
1.1	Índice de texto	1
1.2	Índice de figuras	2
1.3	Índice de tabelas	3
2	Introdução	4
3	Decisões Tomadas.....	5
3.1	Pré-processamento das imagens.....	5
3.2	Targets	6
3.3	Organização dos Dados e Funções.....	7
4	Testes e Resultados.....	8
4.1	Alínea A – Análise com Imagens da Pasta start	8
4.2	Alínea B - Testes com a Totalidade de Imagens da Pasta train.....	10
4.2.1	Conclusão geral da alínea b).....	15
4.3	Alínea C	16
4.3.1	Tarefa 1 - Sem treinar as redes	17
4.3.2	Tarefa 2 - Treinar a rede só com os exemplos da pasta test.....	20
4.3.3	Tarefa 3 - Treinar com todas as imagens (start + train + test).....	25
4.4	Alínea D.....	30
5	Alínea E - Aplicação Gráfica.....	32
6	Conclusões.....	35

1.2 Índice de figuras

Figura 1: Alínea C – Tarefa 1: Matriz de confusão da rede best_net_1 aplicada ao conjunto test sem treino adicional.....	18
Figura 2: Alínea C – Tarefa 1: Matriz de confusão da rede best_net_2 aplicada ao conjunto test sem treino adicional.....	19
Figura 3: Alínea C – Tarefa 1: Matriz de confusão da rede best_net_3 aplicada ao conjunto test sem treino adicional.....	19
Figura 4: Alínea C – Tarefa 2: Matrizes de confusão da rede best_net_1 para as pastas start , test e train , respetivamente	22
Figura 5: Alínea C – Tarefa 2: Matrizes de confusão da rede best_net_2 para as pastas start , test e train , respetivamente	23
Figura 6: Alínea C – Tarefa 2: Matrizes de confusão da rede best_net_3 para as pastas start , test e train , respetivamente	24
Figura 7: Alínea C – Tarefa 2: Matrizes de confusão da rede best_net_1 para as pastas start , test e train , respetivamente	27
Figura 8: Alínea C – Tarefa 2: Matrizes de confusão da rede best_net_2 para as pastas start , test e train , respetivamente	28
Figura 9: Alínea C – Tarefa 2: Matrizes de confusão da rede best_net_3 para as pastas start , test e train , respetivamente	29
Figura 10: Aplicação gráfica – Display geral	32
Figura 11: Aplicação gráfica – Importação de uma rede	32
Figura 12: Aplicação gráfica – Treinamento	33
Figura 13: Aplicação gráfica – Carregamento e classificação de uma imagem	33

1.3 Índice de tabelas

Tabela 1: Codificação one-hot	6
Tabela 2: Organização dos ficheiros.m.....	7
Tabela 3: Diferentes quantidades de camadas e neurónios	9
Tabela 4: Train – Configuração por defeito	10
Tabela 5: Configurações testadas para analisar o impacto do número e dimensão das camadas escondidas na performance da rede neuronal.	11
Tabela 6: Comparação do impacto de diferentes funções de treino no desempenho da rede neuronal.	12
Tabela 7: Comparação entre diferentes funções de ativação usadas nas redes neuronais.....	13
Tabela 8: Impacto de diferentes ratios de divisão dos dados (treino, validação e teste) no desempenho da rede neuronal.	14
Tabela 9: Alínea C - Resultados da Tarefa 1	17
Tabela 10: Alínea C - Resultados da Tarefa 2	20
Tabela 11: Alínea C - Resultados da Tarefa 3	25
Tabela 12: Alínea D - Resultados	30

2 INTRODUÇÃO

Este relatório foi elaborado no âmbito da unidade curricular de **Conhecimento e Raciocínio**, do curso de **Licenciatura em Engenharia Informática**, no Instituto Superior de Engenharia de Coimbra, no ano letivo de 2024/2025.

Este trabalho tem como principal objetivo a **exploração e aprofundamento dos conceitos de redes neuronais do tipo feedforward**, utilizando a linguagem **MATLAB** com o apoio da **toolbox de Deep Learning**. Através da implementação de várias arquiteturas de redes, pretende-se desenvolver classificadores automáticos capazes de **identificar corretamente formas geométricas** a partir de imagens.

Para tal, foi fornecido um conjunto de **390 imagens** a preto e branco, organizadas em três pastas distintas (start, train e test), contendo cada uma 65 imagens de seis classes geométricas: **círculo, papagaio (kite), paralelogramo, quadrado, trapézio e triângulo**.

O trabalho foi dividido em várias fases, conforme definido no enunciado:

- **Conversão e treino inicial com imagens da pasta start**, testando redes simples;
- **Exploração de diferentes topologias, funções de treino e ativação**, usando as imagens da pasta train;
- **Avaliação da generalização das melhores redes**, treinando e testando com os três conjuntos de dados;
- **Classificação de imagens desenhadas manualmente**;
- **Desenvolvimento de uma aplicação gráfica interativa** que permita executar estas tarefas de forma acessível ao utilizador.

Ao longo do relatório serão descritas as decisões de implementação, os testes realizados e as conclusões obtidas, com destaque para os **resultados quantitativos** (precisão global e de teste) e para a **análise das configurações com melhor desempenho**.

3 DECISÕES TOMADAS

Esta secção descreve as escolhas feitas durante a implementação do trabalho, desde o tratamento das imagens até à estruturação dos dados e definição dos targets.

3.1 Pré-processamento das imagens

Para otimizar o desempenho do treino das redes neuronais, as imagens foram sujeitas a um pré-processamento que incluiu os seguintes passos:

- **Redimensionamento:** Todas as imagens foram convertidas para uma resolução padrão de **25x25 píxeis** (original é 224x224), o que permite reduzir o número de entradas da rede e, consequentemente, o tempo de treino, sem perder as características relevantes das formas.
- **Conversão para escala de cinzento:** Caso as imagens contivessem canais RGB (cor), foram convertidas para **tons de cinzento** utilizando a função `rgb2gray` do MATLAB.
- **Binarização:** As imagens foram transformadas em **matrizes binárias** (com valores 0 e 1), usando a função `imbinarize`. A cor preta representa valor 1 (presença de forma) e a branca o valor 0 (fundo), o que facilita o reconhecimento por parte da rede.
- **Vetorização:** Cada imagem binarizada foi transformada numa **coluna de um vetor**, de forma a ficar no formato necessário para ser usada como entrada numa rede feedforward do MATLAB.

Este tratamento foi aplicado a todas as imagens das pastas **start**, **train** e **test**, sendo cada conjunto utilizado conforme definido no enunciado para treino e teste das redes.

3.2 Targets

A saída desejada para cada imagem (target) foi representada utilizando **codificação one-hot** com **6 classes**:

Tabela 1: Codificação one-hot

Classe	Codificação
Círculo	[1 0 0 0 0 0]
Papagaio	[0 1 0 0 0 0]
Paralelogramo	[0 0 1 0 0 0]
Quadrado	[0 0 0 1 0 0]
Trapézio	[0 0 0 0 1 0]
Triângulo	[0 0 0 0 0 1]

Este tipo de representação permite à rede aprender a associar padrões visuais específicos a saídas únicas, facilitando a tarefa de classificação.

3.3 Organização dos Dados e Funções

Para facilitar o desenvolvimento modular do trabalho, foram criadas várias **funções separadas** para o tratamento de cada pasta (start, train, test), bem como para processar imagens individuais (por exemplo, as desenhadas à mão na alínea d).

Tabela 2: Organização dos ficheiros.m

Ficheiro .m	Descrição breve
Pasta “helper”	
saveImagensResized.m	Carrega imagens organizadas por classes, redimensiona para 25x25, binariza e guarda os dados (imagens e targets) em ficheiros .mat.
tratarImagens.m	Igual ao saveImagensResized.m, mas sem guardar ficheiros, prepara os dados (imagens e targets) para uso direto em memória.
trainNeuralNetworks.m	Treina uma rede neural com os dados fornecidos, calcula a precisão de treino e devolve a rede treinada, desempenho e saída simulada.
testNeuralNetworks.m	Avalia a performance da rede com dados de teste, calcula a precisão com base nas previsões e nos targets.
savePlots.m	Gera e guarda um gráfico de matriz de confusão com detalhes da rede no nome do ficheiro da arquitetura da rede e do treino.
writeNetToExcel.m	Exporta para Excel os parâmetros da rede treinada (estrutura, funções, rácios de divisão, nº de épocas) e as precisões obtidas no treino e no teste.
writeMediaAccToExcel.m	Escreve num ficheiro Excel os valores medianos de precisão obtidos durante o treino e teste de uma rede neural, junto com metadados.
Pasta Geral	
funcaoA.m (funcaoA)	Testa múltiplas configurações de rede (neurónios, funções, divisões), regista métricas e guarda as 3 melhores redes.
exci.m	Carrega redes já treinadas, testa com um conjunto (test), e guarda gráficos e métricas.
excii.m	Treina 3 redes, testa com start, train e test, e guarda métricas, gráficos e modelos.
exciii.m	Executa 10 testes por rede e conjunto, calcula médias, gera gráficos e exporta resultados.
exd.m	Testa redes treinadas com imagens desenhadas pelo utilizador, mostrando esperado vs previsão.
program.mlapp	GUI completa que permite configurar, treinar, testar e classificar redes com controlo visual e intuitivo.

4 TESTES E RESULTADOS

4.1 Alínea A – Análise com Imagens da Pasta start

Nesta primeira fase do trabalho, o objetivo principal foi testar o desempenho de redes neuronais *feedforward* simples, utilizando um conjunto restrito de dados, as imagens contidas na pasta **start**, que inclui 5 exemplos de cada uma das 6 classes geométricas.

Inicialmente, procedeu-se ao pré-processamento das imagens com recurso à função **tratarImagens** (ou **saveImagensResized**), que converteu cada imagem para uma matriz binária após redimensionamento (25x25 pixels), permitindo obter as matrizes de entrada (**binaries**) e de saída (**target**) no formato one-hot.

De seguida, foram criadas redes *feedforward* com uma única camada oculta contendo 10 neurónios. Estas redes foram treinadas com todos os dados disponíveis (sem divisão em treino/teste), tal como solicitado, de modo a encontrar configurações capazes de atingir uma precisão global de 100%.

Por fim, mantendo as funções de ativação e treino por omissão (default), foram exploradas diferentes topologias (**com mais neurónios e/ou mais camadas ocultas**) com o objetivo de comparar os desempenhos e avaliar a robustez das redes com estruturas mais complexas.

Todas as configurações testadas, bem como os seus resultados, foram registados automaticamente em ficheiros Excel, facilitando a análise comparativa e o acompanhamento da evolução dos testes.

Tabela 3: Diferentes quantidades de camadas e neurónios

Diferentes quantidades de camadas e neurónios				
Topologias	Função de Treino	Funções de Ativação	Épocas	Melhor Precisão
5	trainlm	tansig / purelin	100	97,67%
10	trainlm	tansig / purelin	100	100%
20	trainlm	tansig / purelin	100	100%
30	trainlm	tansig / purelin	100	100%
10 10	trainlm	tansig / purelin	100	100%
20 20	trainlm	tansig / purelin	100	100%

Os resultados obtidos com as diferentes topologias testadas indicam que, neste cenário limitado da pasta **start**, praticamente todas as redes conseguiram alcançar uma **precisão máxima de 100%**, independentemente do número de neurónios ou camadas utilizadas.

Isto mostra que o conjunto de dados, por ser pequeno e simples, não exige redes complexas para ser corretamente classificado. No entanto, esta precisão elevada pode não ser totalmente verdadeira, pois resulta de um conjunto de treino muito reduzido, o que não garante capacidade de generalização.

A principal diferença observada entre as configurações esteve no **tempo de treino**, que aumentou consideravelmente à medida que se adicionaram mais camadas e neurónios.

Assim, conclui-se que para este conjunto específico de imagens, uma rede simples (por exemplo, 1 camada com 10 neurónios) é suficiente para atingir os objetivos propostos, respondendo positivamente ao que é pedido na alínea a): encontrar uma ou mais parametrizações da rede que atinjam **100% de precisão global** no reconhecimento das imagens da pasta **start**.

4.2 Alínea B - Testes com a Totalidade de Imagens da Pasta train

Nesta segunda fase do trabalho prático, foi realizada uma exploração mais alargada das capacidades das redes neuronais *feedforward*, recorrendo à totalidade das imagens disponibilizadas na pasta **train**. Com um volume de dados significativamente superior ao da alínea anterior, esta etapa permite analisar o comportamento das redes em cenários mais realistas, onde o volume e diversidade dos dados impõem maior exigência à capacidade de generalização das redes.

Foram testadas várias configurações de rede, variando parâmetros fundamentais como o **número de neurónios**, o **número de camadas** ocultas (topologias), as **funções de ativação** e de **treino**, e ainda os **rácios de divisão dos dados em treino, validação e teste**. Esta multiplicidade de testes permitiu avaliar o impacto individual de cada parâmetro no desempenho global e específico da rede.

Todas as configurações foram executadas múltiplas vezes, registando-se automaticamente as precisões globais e de teste, bem como as respetivas matrizes de confusão. As redes com melhor desempenho foram selecionadas com base nesses resultados e posteriormente guardadas para utilização em fases seguintes do trabalho.

Tabela 4: Train – Configuração por defeito

	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos		Precisão Global	Precisão Teste
Configuração por defeito	1	10	tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}		88,77	66,22

Tabela 5: Configurações testadas para analisar o impacto do número e dimensão das camadas escondidas na performance da rede neuronal.

O número e dimensão das camadas escondidas influencia o desempenho?								
	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos		Precisão Global	Precisão Teste
Conf1	1	5	tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}		84,9	60,67
Conf2	1	20	tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}		90,63	70
Conf3	2	5, 5	tansig, tansig, purelin, tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}		84,9	59,78
Conf4	2	10, 10	tansig, tansig, tansig, tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}		88,97	66,22

Com base nos resultados obtidos, observa-se que a variação no número de camadas escondidas e na dimensão de cada camada influenciou de forma **moderada** o desempenho da rede neuronal, tanto em termos de precisão global como na precisão sobre os dados de teste.

A configuração com uma única camada e 20 neurónios (**Conf2**) destacou-se com a **melhor precisão de teste (70%)**, enquanto topologias mais complexas com múltiplas camadas (**Conf3** e **Conf4**) **não mostraram melhorias significativas** face às mais simples.

Adicionalmente, como já verificado na alínea anterior, o tempo de treino **aumenta significativamente** com o número de camadas e neurónios, o que pode representar um custo computacional desnecessário sem ganhos reais de desempenho.

Assim, conclui-se que **redes com topologias simples** (como uma única camada com 20 neurónios) são eficazes para este problema e **apresentam uma boa relação entre desempenho e eficiência** computacional.

Tabela 6: Comparação do impacto de diferentes funções de treino no desempenho da rede neuronal.

A função de treino influencia o desempenho?								
	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos		Precisão Global	Precisão Teste
Conf1	1	10	tansig, purelin	traingd	dividerand = {0.7, 0.15, 0.15}		25,1	21,78
Conf2	1	10	tansig, purelin	trainbfg	dividerand = {0.7, 0.15, 0.15}		43,53	36
Conf3	1	10	tansig, purelin	trainbr	dividerand = {0.7, 0.15, 0.15}		96,17	74,44
Conf4	1	10	tansig, purelin	traincgf	dividerand = {0.7, 0.15, 0.15}		58,07	48,89

Com base nos resultados obtidos, é evidente que a **função de treino** tem um impacto direto e significativo no desempenho das redes neuronais.

A função **trainbr** (Conf3) destacou-se claramente com a **melhor precisão global (96,17%)** e uma **precisão de teste de 74,44%**, demonstrando ser a mais eficaz entre as avaliadas.

Em contraste, funções como **traingd** (Conf1) e **trainbfg** (Conf2) apresentaram desempenhos bastante inferiores, com valores abaixo dos 45% em precisão global e ainda mais reduzidos nos testes. A função **traincgf** (Conf4) teve um desempenho intermédio, mas ainda assim pior que a **trainbr**.

Estes resultados confirmam que **a escolha da função de treino é crítica para a eficácia do modelo**. Embora funções mais robustas como **trainbr** possam exigir mais tempo de treino, os ganhos em desempenho justificam essa escolha, especialmente em problemas com maior volume de dados e necessidade de generalização mais eficaz.

Tabela 7: Comparação entre diferentes funções de ativação usadas nas redes neurais.

As funções de ativação influenciam o desempenho?								
	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos		Precisão Global	Precisão Teste
Conf1	1	10	tansig, logsig	trainlm	dividerand = {0.7, 0.15, 0.15}		72	58,67
Conf2	1	10	purelin, logsig	trainlm	dividerand = {0.7, 0.15, 0.15}		37,47	34,44
Conf3	1	10	logsig, tansig	trainlm	dividerand = {0.7, 0.15, 0.15}		86,87	69,56
Conf4	1	10	radbasn, tansig	trainlm	dividerand = {0.7, 0.15, 0.15}		59,9	34,67

Com os testes realizados, verificou-se que a escolha das **funções de ativação** tem impacto direto no desempenho da rede neuronal.

A configuração **Conf3**, que combina **logsig** na camada escondida com **tansig** na camada de saída, destacou-se com os melhores resultados: **86,87% de precisão global** e **69,56% de precisão de teste**, revelando-se uma combinação eficaz para este problema de classificação.

Por outro lado, configurações como a **Conf2** (**purelin, logsig**) e **Conf4** (**radbasn, tansig**) apresentaram desempenhos consideravelmente mais baixos.

Estes resultados evidenciam que a escolha das funções de ativação é um fator **crítico na modelação da rede**, sendo desejável recorrer a funções como **logsig** e **tansig** que melhor capturam a natureza contínua e não linear dos dados envolvidos na tarefa de classificação de formas geométricas.

Tabela 8: Impacto de diferentes ratios de divisão dos dados (treino, validação e teste) no desempenho da rede neuronal.

A divisão de exemplos pelos conjuntos influencia o desempenho?								
	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos		Precisão Global	Precisão Teste
Conf1	1	10	tansig, purelin	trainlm	dividerand = {0.33, 0.33, 0.33}		75,1	64,2
Conf2	1	10	tansig, purelin	trainlm	dividerand = {0.9, 0.05, 0.05}		95,43	71,33
Conf3	1	10	tansig, purelin	trainlm	dividerand = {0.8, 0.1, 0.1}		92,4	65,33
Conf4	1	10	tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}		86,37	60,22

Com a realização de testes que variaram exclusivamente os rácios de divisão dos dados entre treino, validação e teste, foi possível observar diferenças **relevantes no desempenho** das redes.

A **Configuração 2**, com 90% dos dados alocados ao treino, destacou-se com os **melhores resultados**, atingindo uma **precisão global de 95,43%** e **precisão de teste de 71,33%**.

Por oposição, a **Configuração 1**, com uma divisão equilibrada entre os três conjuntos (33% cada), obteve os piores resultados entre os testados, com apenas **75,1% de precisão global** e **64,2% de precisão de teste**.

As restantes configurações (**Conf3** e **Conf4**) apresentaram desempenhos intermédios, confirmando que **uma maior percentagem de dados reservada para treino tende a beneficiar o desempenho da rede**, desde que ainda haja dados suficientes para validação e teste.

Estes resultados reforçam a importância de ajustar corretamente os rácios de divisão de forma a maximizar a aprendizagem sem comprometer a validação do modelo.

4.2.1 Conclusão geral da alínea b)

Após a realização de vários testes com diferentes configurações de redes neuronais *feedforward*, foi possível avaliar o impacto individual de quatro parâmetros fundamentais no desempenho das redes: **topologia**, **função de treino**, **funções de ativação** e **divisão dos dados**.

Em relação à **topologia**, concluiu-se que **redes simples** (como uma única camada com 20 neurónios) já são capazes de atingir bons resultados, sendo que o aumento do número de camadas ou neurónios não trouxe melhorias significativas e apenas aumentou o custo computacional.

A análise das **funções de treino** revelou ser este o parâmetro com **maior impacto direto** na performance. A função **trainbr** destacou-se consistentemente com as melhores precisões, demonstrando ser uma escolha robusta para este tipo de tarefa, mesmo que exija mais tempo de treino.

Quanto às **funções de ativação**, notou-se que combinações com funções como **logsig** e **tansig** apresentaram **desempenhos superiores**, enquanto outras mais lineares ou não convencionais, como **purelin** ou **radbasn**, reduziram significativamente a capacidade de generalização da rede.

Por fim, no que respeita à **divisão dos dados**, observou-se que alocar uma **maior percentagem para treino (por exemplo, 90%)** contribuiu para melhores desempenhos, desde que acompanhada por conjuntos mínimos de validação e teste que permitam uma avaliação fiável.

Com base nestas análises, foram guardadas as **três redes com melhor desempenho** geral para uso nas fases seguintes do trabalho. Estes testes permitiram consolidar práticas eficazes de configuração e treino de redes neuronais, que serão essenciais na generalização e avaliação com novos dados nas alíneas seguintes.

4.3 Alínea C

Nesta fase do trabalho, procurou-se testar a **capacidade de generalização** das redes neuronais desenvolvidas, alargando a análise aos três conjuntos de dados disponíveis: start, train e test. O objetivo principal foi observar o comportamento das redes quando treinadas e/ou testadas em cenários distintos, simulando contextos mais próximos da aplicação real.

Para isso, foram reutilizadas as **três melhores redes previamente treinadas** na alínea b), e aplicadas a dados que não fizeram parte do seu treino inicial, permitindo avaliar o grau de adaptação a novas amostras. Foram também realizados testes complementares onde as redes foram **treinadas apenas com a pasta test** e, por fim, **com todos os dados combinados** (start + train + test), de modo a analisar o efeito de um treino mais completo sobre a precisão de classificação.

Em todas as abordagens, foram recolhidas as **precisões globais e de teste**, bem como as **matrizes de confusão**, com o objetivo de identificar padrões de erro e analisar o desempenho por classe. Esta alínea é, assim, fundamental para validar a robustez dos modelos e fundamentar a escolha final das redes a aplicar em novos cenários.

4.3.1 Tarefa 1 - Sem treinar as redes

Nesta primeira etapa, as redes previamente gravadas foram aplicadas diretamente às imagens da pasta **test**, **sem qualquer novo treino**, permitindo avaliar a sua robustez face a dados não vistos. Foram recolhidas as precisões de teste e as matrizes de confusão, comparando os resultados com os obtidos durante a alínea b).

Tabela 9: Alínea C - Resultados da Tarefa 1

Melhor rede usada	Precisão Global (%)
best_net_1	71,67
best_net_2	100
best_net_3	100

As redes **best_net_2** e **best_net_3** mantêm um **desempenho perfeito**, classificando corretamente todas as imagens, mesmo sem novo treino. Isto demonstra que foram **bem otimizadas na alínea b)** e que conseguem **reter o conhecimento adquirido com eficácia**.

Já a rede **best_net_1** caiu para **71,67% de precisão**, revelando uma **menor capacidade de generalização** e maior sensibilidade à ausência de novo ajuste. Este valor sugere que esta rede depende fortemente do contexto em que foi treinada.

Estas diferenças indicam que **as redes 2 e 3 estão mais bem adaptadas a problemas de classificação estáticos**, onde se espera desempenho imediato sem reproprocessamento.

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	6	
1	10 16.7%	1 1.7%	0 0.0%	0 0.0%	2 3.3%	0 0.0%	76.9% 23.1%
2	0 0.0%	5 8.3%	0 0.0%	0 0.0%	0 0.0%	1 1.7%	83.3% 16.7%
3	0 0.0%	2 3.3%	8 13.3%	1 1.7%	2 3.3%	1 1.7%	57.1% 42.9%
4	0 0.0%	2 3.3%	2 3.3%	9 15.0%	1 1.7%	2 3.3%	56.2% 43.8%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 8.3%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 10.0%	100% 0.0%
	100% 0.0%	50.0% 50.0%	80.0% 20.0%	90.0% 10.0%	50.0% 50.0%	60.0% 40.0%	71.7% 28.3%

Cada célula representa a quantidade de classificações feitas pela rede (em número absoluto e percentagem). As diagonais (a verde) indicam classificações corretas, enquanto as restantes células representam erros de classificação. Os valores à direita e na parte inferior da matriz mostram a **precisão por classe** (vertical: output; horizontal: target).

Na matriz de confusão:

- Linha/columa 1 → Circle
- Linha/columa 2 → Kite
- Linha/columa 3 → Parallelogram
- Linha/columa 4 → Square
- Linha/columa 5 → Trapezoid
- Linha/columa 6 → Triangle

Figura 1: Alínea C – Tarefa 1: Matriz de confusão da rede **best_net_1** aplicada ao conjunto test sem treino adicional.

Precisão global: 71,7%

A rede mostra um desempenho muito desigual entre classes:

- **Classe 1 (Circle)** foi corretamente classificada com **100% de acerto**.
- **Classe 5 (Trapezoid)** e **classe 6 (Triangle)** atingem **50% e 60% de precisão, respetivamente**.
- **Classe 2 (Kite)** tem apenas **50% de acerto**, com erros significativos (principalmente confundida com classes 4 e 3).
- **Classe 3 (Parallelogram)** alcança **80%**, mas ainda tem erros dispersos nas classes 2, 4 e 5.
- **Classe 4 (Square)** apresenta **90% de acerto**, com pequenas confusões para as classes 3, 5 e 6.

Apesar de alguma estrutura de acerto nas diagonais, há **muitas confusões cruzadas** e distribuição de erros por quase todas as classes.

A rede parece **não ter mantido bem o conhecimento** quando usada diretamente sem re-treino, o que é consistente com a **queda acentuada da precisão global** face às outras redes.

Confusion Matrix							
Output Class	1	2	3	4	5	6	
	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	100% 0.0%
1	2	3	4	5	6		
							100% 0.0%

Figura 2: Alínea C – Tarefa 1: Matriz de confusão da rede **best_net_2** aplicada ao conjunto test sem treino adicional.

Precisão global: 100%

A rede apresentou um **desempenho perfeito**, classificando corretamente todas as imagens da pasta test sem qualquer erro.

Todas as classes obtiveram **100% de acerto**, com acertos alinhados na diagonal da matriz de confusão e **zero confusões cruzadas**.

Este resultado demonstra que a **best_net_2** tem uma **memorização forte e estável** das características dos dados com que foi treinada.

Apesar disso, é importante lembrar que este tipo de desempenho pode também **indicar sobreajuste**, isto é, a rede aprendeu tão bem os dados de treino que **não necessariamente generaliza bem para novos dados** (como foi analisado nas tarefas seguintes).

Confusion Matrix							
Output Class	1	2	3	4	5	6	
	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	100% 0.0%
	1	2	3	4	5	6	
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

Figura 3: Alínea C – Tarefa 1: Matriz de confusão da rede **best_net_3** aplicada ao conjunto test sem treino adicional.

Precisão global: 100%

Tal como a **best_net_2**, esta rede classificou **corretamente todas as amostras**, obtendo **100% de acerto em todas as classes**.

A matriz mostra uma **diagonal perfeitamente preenchida**, sem qualquer erro fora dela, não havendo confusões entre classes.

O desempenho confirma que a rede foi **treinada de forma eficaz e assimilou bem os dados da pasta test**, mesmo sem re-treino.

Ainda assim, este tipo de performance pode também indicar **forte memorização dos dados de treino**, o que, como vimos nas tarefas seguintes, **não se traduz necessariamente em boa generalização**.

4.3.2 Tarefa 2 - Treinar a rede só com os exemplos da pasta test

Nesta tarefa, o objetivo foi testar a capacidade das redes neuronais de **aprender a partir de um conjunto de dados reduzido e isolado**, neste caso, as imagens contidas exclusivamente na pasta test. As três melhores redes selecionadas na alínea b) foram agora **reconfiguradas e treinadas de raiz apenas com estes dados**, permitindo observar se o treino localizado melhora a sua precisão face a novos testes.

Após o treino, cada rede foi utilizada para classificar os dados dos três conjuntos (start, train e test), recolhendo-se as **precisões de teste em cada caso**, bem como as respetivas **matrizes de confusão**. Esta abordagem permitiu perceber **até que ponto um conjunto de treino limitado compromete (ou não) a capacidade de generalização**, especialmente quando a rede é confrontada com dados que não foram incluídos no treino.

A análise desta tarefa é particularmente relevante, uma vez que simula cenários onde o modelo é treinado em condições restritas, o que pode ocorrer em problemas do mundo real com acesso limitado a dados anotados.

Tabela 10: Alínea C - Resultados da Tarefa 2

Melhor Rede Usada	Pasta Usada	Precisão Global (%)
best_net_1	Start	93,3
	Train	96,7
	Test	84,7
best_net_2	Start	60
	Train	100
	Test	77,7
best_net_3	Start	70
	Train	100
	Test	79

A Tabela 10 mostra que, embora as redes tenham alcançado 100% de precisão no conjunto de treino (test), o seu desempenho caiu significativamente quando aplicadas a dados diferentes (start e train), revelando fraca capacidade de generalização.

A exceção foi a **best_net_1**, que manteve resultados elevados nos três conjuntos, indicando maior robustez.

Estes resultados demonstram que treinar uma rede apenas com dados limitados compromete a sua eficácia em cenários mais amplos, realçando a importância da diversidade no conjunto de treino.

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	6	
1	5 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	4 13.3%	0 0.0%	0 0.0%	1 3.3%	0 0.0%	80.0% 20.0%
3	0 0.0%	0 0.0%	5 16.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	5 16.7%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 13.3%	0 0.0%	100% 0.0%
6	0 0.0%	1 3.3%	0 0.0%	0 0.0%	0 0.0%	5 16.7%	83.3% 16.7%
	100% 0.0%	80.0% 20.0%	100% 0.0%	100% 0.0%	80.0% 20.0%	100% 0.0%	93.3% 6.7%

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	6	
1	9 15.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	10 16.7%	1 1.7%	0 0.0%	90.9% 9.1%
5	1 1.7%	0 0.0%	0 0.0%	0 0.0%	9 15.0%	0 0.0%	90.0% 10.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	100% 0.0%
	90.0% 10.0%	100% 0.0%	100% 0.0%	100% 0.0%	90.0% 10.0%	100% 0.0%	96.7% 3.3%

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	6	
1	36 12.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	41 13.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	2 0.7%	42 14.0%	6 2.0%	1 0.3%	4 1.3%	76.4% 23.6%
4	3 1.0%	0 0.0%	5 1.7%	42 14.0%	1 0.3%	0 0.0%	82.4% 17.6%
5	10 3.3%	0 0.0%	3 1.0%	1 0.3%	47 15.7%	0 0.0%	77.0% 23.0%
6	1 0.3%	7 2.3%	0 0.0%	1 0.3%	1 0.3%	46 15.3%	82.1% 17.9%
	72.0% 28.0%	82.0% 18.0%	84.0% 16.0%	84.0% 16.0%	94.0% 6.0%	92.0% 8.0%	84.7% 15.3%

Figura 4: Alínea C – Tarefa 2: Matrizes de confusão da rede **best_net_1** para as pastas **start**, **test** e **train**, respetivamente

Pasta start

- **Precisão global: 93,3%**
- Muito bom desempenho em generalização: 4 classes com **100% de acerto**.
- Classes 2 (*Kite*) e 6 (*Triangle*) tiveram confusões entre si, mas com impacto limitado.
- Demonstra que a rede **transfere bem o conhecimento aprendido para dados visuais diferentes**.

Pasta test

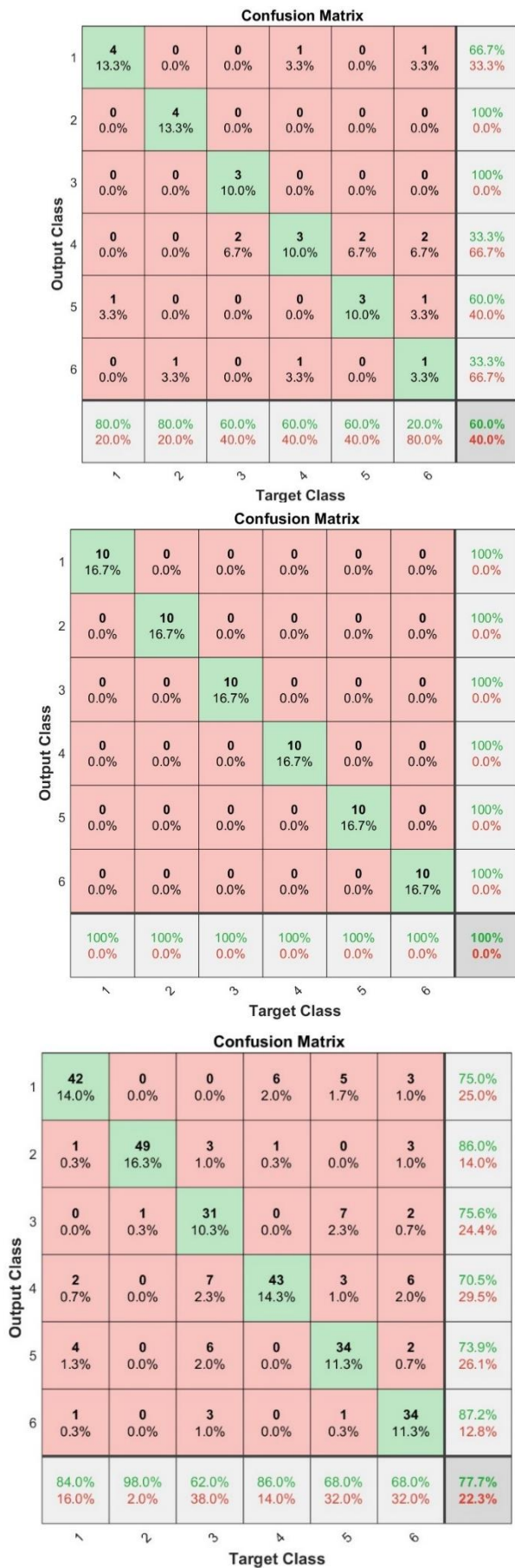
- **Precisão global: 96,7%**
- Classificação excelente, com **100% de acerto** em 4 das 6 classes.
- Erros mínimos e pontuais nas classes 1 e 5.
- Mostra **forte aprendizagem dos dados de treino**, sem sinais preocupantes de sobreajuste.

Pasta train

- **Precisão global: 84,7%**
- Desempenho mais irregular:
 - Classe 1 (*Circle*) com apenas **72% de precisão**, com erros dispersos.
 - As restantes variaram entre **77% e 94%**, mantendo-se aceitáveis.
- A matriz mostra que, embora a rede generalize razoavelmente, **formas com mais variação (como círculos)** são mais difíceis de classificar.

A rede **best_net_1** apresenta um **desempenho muito sólido e equilibrado**:

- Excelente no seu próprio conjunto (test).
- **Boa generalização para start**, e **aceitável para train**, com algumas classes mais difíceis.
- É a **mais estável e confiável** entre as três redes analisadas, mostrando que aprendeu **padrões relevantes e reutilizáveis**.



Pasta start

- **Precisão global: 60,0%**
- Classes 1 (*Circle*) e 2 (*Kite*) tiveram desempenho bom com poucos erros.
- A rede apresenta resultados razoáveis em quase todas as classes, com exceção da classe 5. A precisão por classe mostra que a capacidade de reconhecimento varia bastante, o que indica instabilidade na generalização do modelo.

Pasta test

- **Precisão global: 100%**
- A rede teve **classificação perfeita** no conjunto com que foi treinada.
- Todas as classes foram reconhecidas sem erros.
- Mostra uma aprendizagem eficaz, mas também sugere **potencial sobreajuste**.

Pasta train

- **Precisão global: 77,7%**
- **Bom desempenho global** da rede.
- **Classe 2 (*Kite*)** destaca-se com desempenho quase perfeito (98%).
- **Classes 1 (*Circle*) e 4 (*Square*)** com precisão elevada (acima de 80%).
- **Classes 3 (*Parallelogram*), 5 (*Trapezoid*) e 6 (*Triangle*)** com recall mais baixo, havendo presença de erros dispersos.
- Indícios de **confusões entre classes visualmente semelhantes**.
- Apesar da boa precisão global (77,7%), há **espaço para melhorar nas classes mais fracas**.

A `best_net_2` revelou um perfil **marcadamente dependente do treino**, com excelente desempenho no conjunto test, mas **desempenho fraco a moderado**

Figura 5: Alínea C – Tarefa 2: Matrizes de confusão da rede `best_net_2` para as pastas **start**, **test** e **train**, respetivamente

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	6	Precisao	Recall
1	5 16.7%	0 0.0%	0 0.0%	2 6.7%	2 6.7%	1 3.3%	50.0%	50.0%
2	0 0.0%	4 13.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%	0.0%
3	0 0.0%	0 0.0%	3 10.0%	0 0.0%	0 0.0%	0 0.0%	100%	0.0%
4	0 0.0%	0 0.0%	2 6.7%	3 10.0%	0 0.0%	1 3.3%	50.0%	50.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 10.0%	0 0.0%	100%	0.0%
6	0 0.0%	1 3.3%	0 0.0%	0 0.0%	0 0.0%	3 10.0%	75.0%	25.0%
	100%	80.0%	60.0%	60.0%	60.0%	60.0%	70.0%	30.0%

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	6	Precisao	Recall
1	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%	0.0%
2	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%	0.0%
3	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	100%	0.0%
4	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	100%	0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	100%	0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	100%	0.0%
	100%	100%	100%	100%	100%	100%	100%	0.0%

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	6	Precisao	Recall
1	41 13.7%	1 0.3%	1 0.3%	6 2.0%	19 6.3%	1 0.3%	59.4%	40.6%
2	0 0.0%	48 16.0%	2 0.7%	1 0.3%	0 0.0%	2 0.7%	90.6%	9.4%
3	0 0.0%	1 0.3%	33 11.0%	0 0.0%	1 0.3%	0 0.0%	94.3%	5.7%
4	4 1.3%	0 0.0%	3 1.0%	43 14.3%	3 1.0%	0 0.0%	81.1%	18.9%
5	5 1.7%	0 0.0%	8 2.7%	0 0.0%	27 9.0%	2 0.7%	64.3%	35.7%
6	0 0.0%	0 0.0%	3 1.0%	0 0.0%	0 0.0%	45 15.0%	93.8%	6.2%
	82.0%	96.0%	66.0%	86.0%	54.0%	90.0%	79.0%	21.0%

Pasta start

- **Precisão global: 70%**
- **Desempenho razoável** da rede.
- **Classes 1 (Circle) e 2 (Kite)** bem reconhecidas, com precisão elevada.
- **Classes 4 (Square), 5 (Trapezoid) e 6 (Triangle)** com **confusões significativas**.

Pasta test

- **Precisão global: 100%**
- Classificação perfeita em todas as classes.
- Como era o conjunto de treino, este resultado confirma que a rede **memoriza eficazmente** os dados, mas não indica capacidade de generalização.

Pasta train

- **Precisão global: 79,0%**
- A rede tem **bom desempenho geral**, especialmente nas classes 2, 4 e 6.
- A **classe 5 é o ponto fraco**, com apenas 54% de recall.
- **Classe 3** também mostra fragilidade moderada.
- A **precisão por classe é maioritariamente acima de 80%**, o que mostra **boa capacidade de generalização**, com **algumas classes a precisar de reforço** (ex: classe 5).

A rede `best_net_3` tem excelente desempenho no seu conjunto de treino (test), mas a sua **capacidade de generalização é moderada**. Os resultados com start e train mostram que a rede é sensível a variações visuais, especialmente em classes com maior diversidade de representação. Embora seja funcional, **não é a mais estável** entre as três redes testadas.

Figura 6: Alínea C – Tarefa 2: Matrizes de confusão da rede `best_net_3` para as pastas **start**, **test** e **train**, respetivamente

4.3.3 Tarefa 3 - Treinar com todas as imagens (start + train + test)

Nesta última tarefa da alínea c), procurou-se maximizar o conhecimento das redes neuronais através do treino com **todo o conjunto de dados disponível**, reunindo as imagens das pastas start, train e test. O objetivo foi perceber se, ao expor a rede a uma amostra mais ampla e diversificada, seria possível melhorar o desempenho global e a capacidade de generalização do modelo.

As melhores redes identificadas na alínea b) foram novamente utilizadas, desta vez treinadas com o conjunto total de dados. Após o treino, cada rede foi testada individualmente em cada uma das três pastas (start, train e test), permitindo avaliar o desempenho da rede em diferentes contextos de teste e verificar se havia alguma variação significativa na precisão de acerto entre os conjuntos.

Foram recolhidas as **precisões de teste por pasta**, bem como as respetivas **matrizes de confusão**, que permitiram identificar **quais as formas geométricas com melhores e piores desempenhos** após o treino alargado. Esta tarefa é especialmente relevante para validar a robustez final das redes e selecionar as mais equilibradas para futuras aplicações.

Tabela 11: Alínea C - Resultados da Tarefa 3

Melhor Rede Usada	Pasta Usada	Precisão Global (%)
best_net_1	Start	93,33
	Train	98,33
	Test	91
best_net_2	Start	70
	Train	100
	Test	86,67
best_net_3	Start	66,67
	Train	100
	Test	72,67

Com base na Tabela 11, pode concluir-se que o treino com todas as imagens disponíveis (start + train + test) levou a um **aumento claro da precisão global** e a uma melhoria da **generalização das redes**.

A **best_net_1** destacou-se com resultados consistentes e elevados em todos os conjuntos (acima de 91%), sendo a mais equilibrada.

Já as redes **best_net_2** e **best_net_3**, apesar de alcançarem 100% no treino, apresentaram **quedas mais acentuadas no conjunto de teste**, refletindo menor robustez.

Estes resultados confirmam que o treino com dados mais diversos contribui para **redes mais fiáveis e adaptáveis a diferentes contextos de teste**.

Confusion Matrix

Output Class	1	2	3	4	5	6	
1	5 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	4 13.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	4 13.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	1 3.3%	1 3.3%	5 16.7%	0 0.0%	0 0.0%	71.4% 28.6%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 16.7%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 16.7%	100% 0.0%
	1	2	3	4	5	6	
	100% 0.0%	80.0% 20.0%	80.0% 20.0%	100% 0.0%	100% 0.0%	100% 0.0%	93.3% 6.7%

Target Class

Confusion Matrix

Output Class	1	2	3	4	5	6	
1	9 15.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	100% 0.0%
5	1 1.7%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	90.9% 9.1%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	100% 0.0%
	1	2	3	4	5	6	
	90.0% 10.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	98.3% 1.7%

Target Class

Confusion Matrix

Output Class	1	2	3	4	5	6	
1	39 13.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	48 16.0%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	98.0% 2.0%
3	0 0.0%	1 0.3%	47 15.7%	6 2.0%	0 0.0%	0 0.0%	87.0% 13.0%
4	5 1.7%	0 0.0%	2 0.7%	44 14.7%	2 0.7%	1 0.3%	81.5% 18.5%
5	6 2.0%	1 0.3%	0 0.0%	0 0.0%	48 16.0%	1 0.3%	85.7% 14.3%
6	0 0.0%	0 0.0%	1 0.3%	0 0.0%	0 0.0%	47 15.7%	97.9% 2.1%
	1	2	3	4	5	6	
	78.0% 22.0%	96.0% 4.0%	94.0% 6.0%	88.0% 12.0%	96.0% 4.0%	94.0% 6.0%	91.0% 9.0%

Target Class

Pasta start

- **Precisão global: 93,3%**
- As Classes 2 e 3, com **80% de recall**, indicando uma pequena margem de melhoria.
- A **precisão global** mostra que a rede reconhece bem os padrões reais das classes, com **forte capacidade de generalização**.

Pasta test

- **Precisão global: 98,3%**
- Desempenho quase perfeito.
- O único erro foi um caso da **classe 1**.
- As restantes classes foram reconhecidas com **100% de precisão**, reforçando a **consistência da aprendizagem**.

Pasta train

- **Precisão global: 91,0%**
- Bom desempenho com a maioria das classes acima de **85% de precisão**.
- Apenas a **classe 1 (Circle)** ficou ligeiramente abaixo, com **78% de acerto**.
- A rede mostra **boa cobertura e generalização**, mesmo num conjunto grande e variado.

A best_net_1, quando treinada com todos os dados disponíveis, demonstrou um comportamento **extremamente sólido e equilibrado**:

- Obteve **precisões globais acima de 90%** em todos os conjuntos.
- Foi **consistente entre classes**, com poucos erros graves e excelente reconhecimento das formas geométricas.
- Mostra-se como uma **rede robusta e altamente generalizável**, ideal para contextos com variabilidade nos dados.

Figura 7: Alínea C – Tarefa 2: Matrizes de confusão da rede **best_net_1** para as pastas **start**, **test** e **train**, respetivamente

Confusion Matrix

Output Class	1	2	3	4	5	6	
1	5 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	3 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	1 3.3%	3 10.0%	0 0.0%	1 3.3%	1 3.3%	50.0% 50.0%
4	0 0.0%	1 3.3%	0 0.0%	3 10.0%	0 0.0%	1 3.3%	60.0% 40.0%
5	0 0.0%	0 0.0%	0 0.0%	2 6.7%	4 13.3%	0 0.0%	66.7% 33.3%
6	0 0.0%	0 0.0%	2 6.7%	0 0.0%	0 0.0%	3 10.0%	60.0% 40.0%
	100% 0.0%	60.0% 40.0%	60.0% 40.0%	60.0% 40.0%	80.0% 20.0%	60.0% 40.0%	70.0% 30.0%
Target Class	1	2	3	4	5	6	

Confusion Matrix

Output Class	1	2	3	4	5	6	
1	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
Target Class	1	2	3	4	5	6	

Confusion Matrix

Output Class	1	2	3	4	5	6	
1	49 16.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	47 15.7%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	97.9% 2.1%
3	0 0.0%	0 0.0%	37 12.3%	0 0.0%	1 0.3%	0 0.0%	97.4% 2.6%
4	1 0.3%	2 0.7%	2 0.7%	38 12.7%	7 2.3%	2 0.7%	73.1% 26.9%
5	0 0.0%	0 0.0%	7 2.3%	11 3.7%	42 14.0%	0 0.0%	70.0% 30.0%
6	0 0.0%	1 0.3%	4 1.3%	1 0.3%	0 0.0%	47 15.7%	88.7% 11.3%
	98.0% 2.0%	94.0% 6.0%	74.0% 26.0%	76.0% 24.0%	84.0% 16.0%	94.0% 6.0%	86.7% 13.3%
Target Class	1	2	3	4	5	6	

Pasta start

- **Precisão global: 70,0%**
- Apenas a classe 1 (*Circle*) teve 100% de acerto.
- As restantes apresentaram entre **50% e 66%**, com erros dispersos.
- A rede teve **dificuldade em aplicar o conhecimento aprendido**, o que evidencia **fragilidade na generalização** para este subconjunto.

Pasta test

- **Precisão global: 100%**
- Classificação perfeita: todas as classes com 100% de acerto.
- Demonstra **boa retenção dos dados de treino**, mas também levanta **sinais de sobreajuste**, dado que os dados estavam no treino.

Pasta train

- **Precisão global: 86,7%**
- Desempenho equilibrado, com classes como:
 - **Classe 1 (98%)**, 2 (94%) e 6 (94%) muito bem classificadas.
 - Outras como **classe 3 (74%)** e **classe 4 (76%)** com maior margem de erro.
- A rede mostra **alguma robustez**, mas a precisão varia significativamente entre formas.

A **best_net_2** apresenta um **desempenho satisfatório em treino e no conjunto test**, mas **sufre quedas notáveis na generalização**, especialmente no conjunto start. Ainda que tenha melhorado face à Tarefa 2, **continua a mostrar inconsistência em classes mais complexas** e é superada por outras redes em termos de estabilidade.

Figura 8: Alínea C – Tarefa 2: Matrizes de confusão da rede **best_net_2** para as pastas **start**, **test** e **train**, respetivamente

Confusion Matrix								
Output Class	1	5 16.7%	0 0.0%	0 0.0%	2 6.7%	1 3.3%	0 0.0%	62.5% 37.5%
	2	0 0.0%	5 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	3 10.0%	0 0.0%	1 3.3%	0 0.0%	75.0% 25.0%
	4	0 0.0%	0 0.0%	2 6.7%	1 3.3%	0 0.0%	0 0.0%	33.3% 66.7%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 6.7%	1 3.3%	66.7% 33.3%
	6	0 0.0%	0 0.0%	0 0.0%	2 6.7%	1 3.3%	4 13.3%	57.1% 42.9%
		100% 0.0%	100% 0.0%	60.0% 40.0%	20.0% 80.0%	40.0% 60.0%	80.0% 20.0%	66.7% 33.3%
		1	2	3	4	5	6	

		Target Class Confusion Matrix						
Output Class	1	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	100% 0.0%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	100% 0.0%
		100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
		Target Class						
		1	2	3	4	5	6	

		Confusion Matrix						
Output Class	1	43 14.3%	0 0.0%	2 0.7%	3 1.0%	12 4.0%	1 0.3%	70.5% 29.5%
	2	4 1.3%	49 16.3%	0 0.0%	0 0.0%	0 0.0%	3 1.0%	87.5% 12.5%
	3	0 0.0%	0 0.0%	27 9.0%	0 0.0%	13 4.3%	2 0.7%	64.3% 35.7%
	4	1 0.3%	0 0.0%	5 1.7%	41 13.7%	9 3.0%	0 0.0%	73.2% 26.8%
	5	0 0.0%	0 0.0%	6 2.0%	0 0.0%	16 5.3%	2 0.7%	66.7% 33.3%
	6	2 0.7%	1 0.3%	10 3.3%	6 2.0%	0 0.0%	42 14.0%	68.9% 31.1%
		86.0% 14.0%	98.0% 2.0%	54.0% 46.0%	82.0% 18.0%	32.0% 68.0%	84.0% 16.0%	72.7% 27.3%
		Target Class						
		1	2	3	4	5	6	

Pasta start

- **Precisão global: 66,7%**
- As únicas classes com 100% de acerto foram **1** (*Circle*) e **2** (*Kite*).
- Existem erros generalizados e dispersos revelam **baixa capacidade de generalização**, mesmo com exposição total ao conjunto.

Pasta test

- **Precisão global: 100%**
- Resultado perfeito, com **todas as classes corretamente classificadas**.
- A rede **memoriza eficazmente os dados**, mas isso não garante robustez fora do conjunto de treino.

Pasta train

- **Precisão global: 72,7%**
- Desempenho globalmente fraco:
 - Classe 5 (*Trapezoid*) teve **apenas 32%**, e classe 3 apenas **54%**.
 - Erros amplamente distribuídos, com sobreposição significativa entre classes.
- A rede demonstra **grandes dificuldades em classes com maior variação visual**, mesmo estando incluídas no treino.

A best_net_3 apresenta um desempenho **muito dependente do conjunto de treino**, com **grandes limitações na generalização**. Apesar de memorizar bem (test a 100%), os resultados com start e train mostram **fragilidade estrutural e confusões recorrentes**. Em contextos práticos, seria a **menos confiável das três redes analisadas na Tarefa 3**.

Figura 9: Alínea C – Tarefa 2: Matrizes de confusão da rede **best_net_3** para as pastas **start**, **test** e **train**, respetivamente

4.4 Alínea D

Nesta alínea, pretende-se avaliar a capacidade da rede neuronal treinada em generalizar para novos exemplos desenhados manualmente ou com recurso a uma ferramenta de desenho. Para isso, foram criadas **5 imagens de cada categoria**. Estas imagens serão convertidas para matrizes binárias e, posteriormente, classificadas pelas melhores redes obtidas anteriormente (na alínea c iv)). O objetivo é verificar a robustez do modelo perante dados novos e semelhantes aos do conjunto de treino, analisando os resultados e tirando conclusões sobre o seu desempenho.

Tabela 12: Alínea D - Resultados

Figura da Pasta	Figura Obtida		
	best_net_1	best_net_2	best_net_3
circle	triangle	triangle	triangle
circle	triangle	square	triangle
circle	square	triangle	circle
circle	square	parallelogram	square
circle	square	triangle	square
kite	triangle	triangle	triangle
kite	triangle	square	triangle
kite	square	triangle	circle
kite	square	parallelogram	square
kite	square	triangle	square
parallelogram	triangle	triangle	triangle
parallelogram	triangle	square	triangle
parallelogram	square	triangle	circle
parallelogram	square	parallelogram	square
parallelogram	square	triangle	square
square	triangle	triangle	triangle

square	triangle	square	triangle
square	square	triangle	circle
square	square	parallelogram	square
square	square	triangle	square
trapezoid	triangle	triangle	triangle
trapezoid	triangle	square	triangle
trapezoid	square	triangle	circle
trapezoid	square	parallelogram	square
trapezoid	square	triangle	square
triangle	triangle	triangle	triangle
triangle	triangle	square	triangle
triangle	square	triangle	circle
triangle	square	parallelogram	square
triangle	square	triangle	square
Precisão Global:	16,67 %	16,67 %	13,33 %

Após aplicar as três melhores redes neurais às imagens criadas manualmente, observou-se uma baixa precisão: **16,67% para best_net_1 e best_net_2, e 13,33% para best_net_3**. Isto evidencia dificuldades na generalização, mesmo com figuras semelhantes às de treino. A maioria das imagens foi incorretamente classificada como “triangle”, revelando um viés do modelo, com poucas previsões corretas, o que mostra fragilidades fora do conjunto de treino. Este resultado pode estar relacionado com fatores como:

- Baixa variabilidade no conjunto de treino original;
- Sobretreino da rede para os exemplos usados;
- Diferenças na espessura, alinhamento ou proporção das figuras desenhadas manualmente;
- Pré-processamento inadequado (por exemplo, binarização imperfeita ou ruído nos dados).

5 ALÍNEA E - APLICAÇÃO GRÁFICA

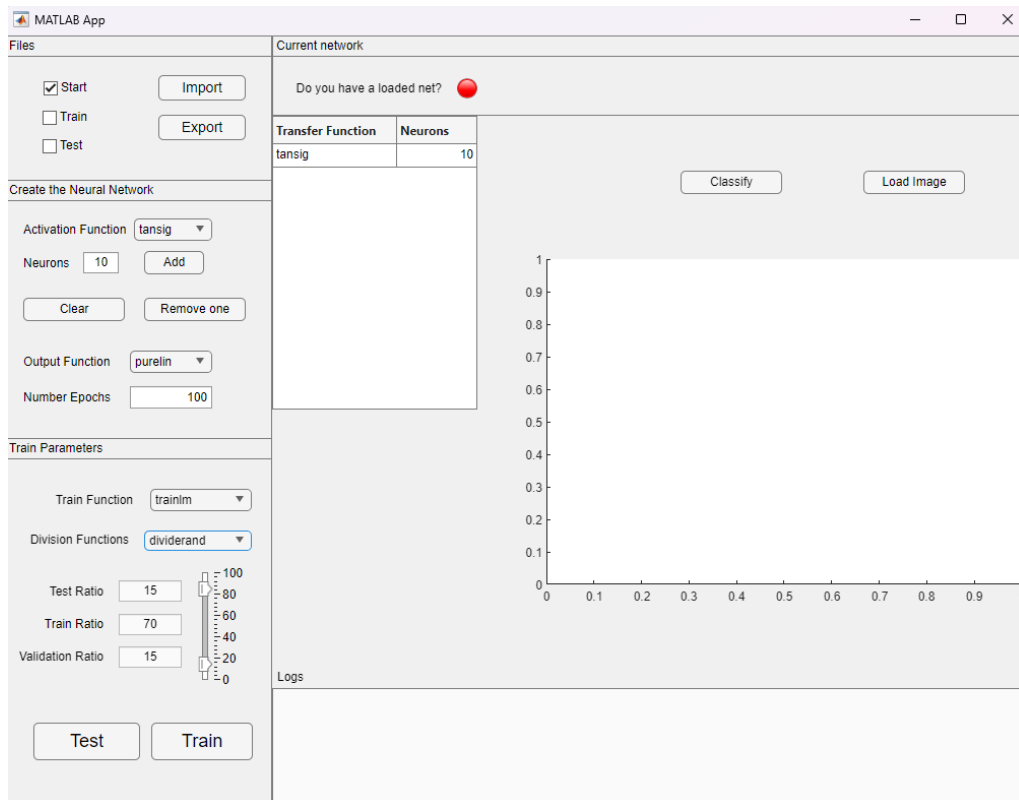


Figura 10: Aplicação gráfica – Display geral

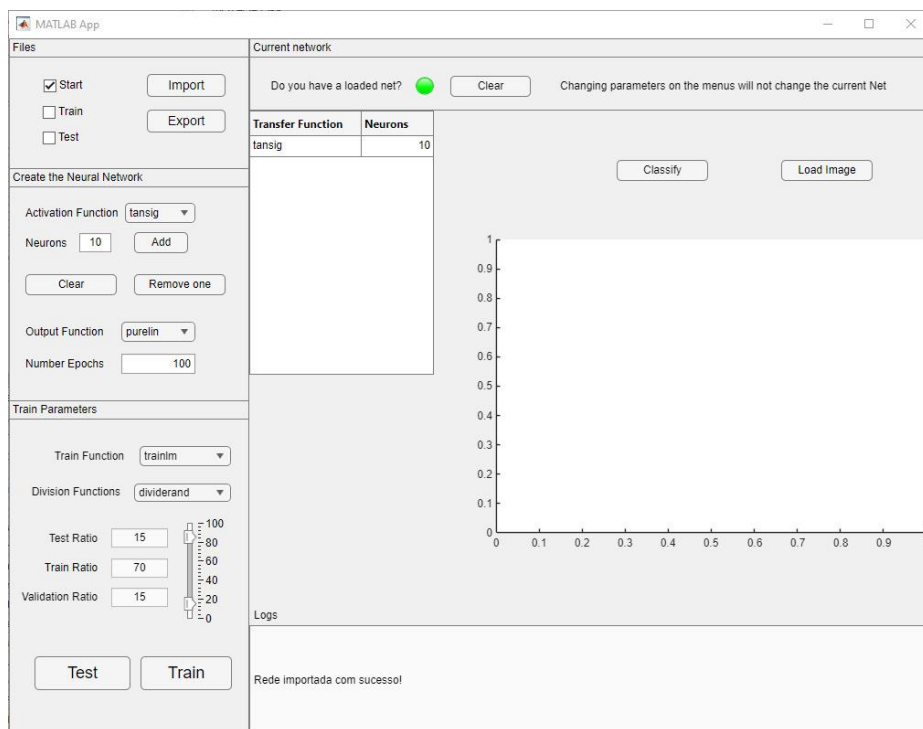


Figura 11: Aplicação gráfica – Importação de uma rede

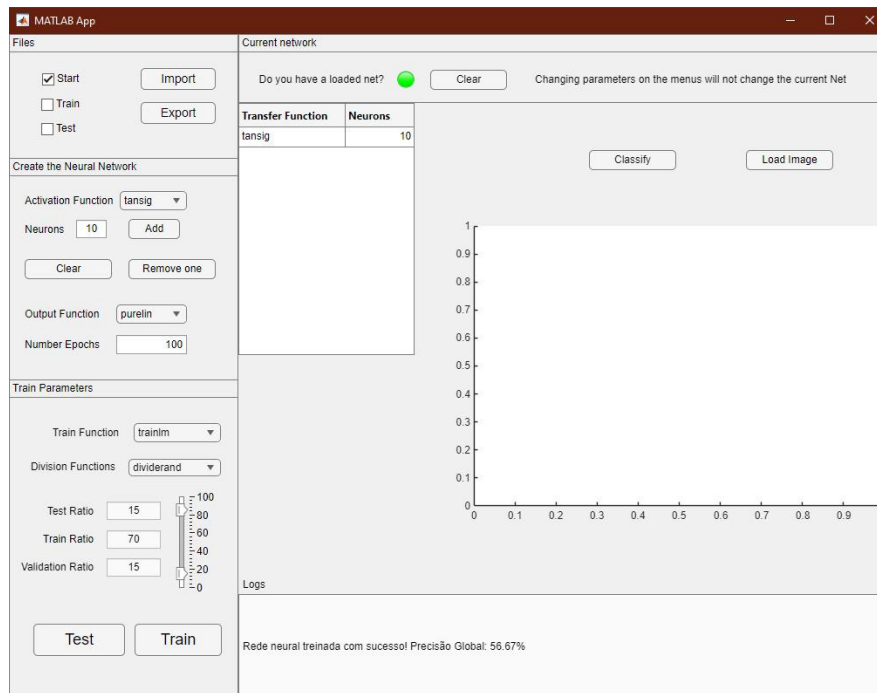


Figura 12: Aplicação gráfica – Treinamento

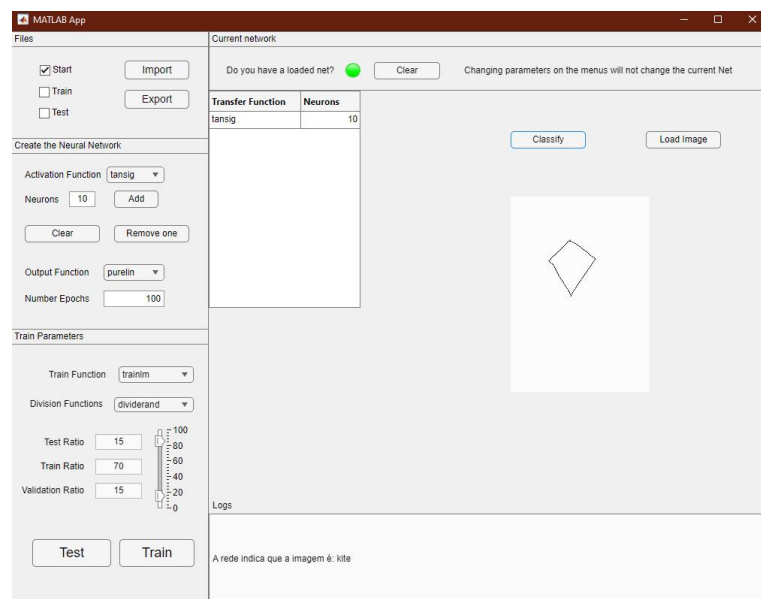
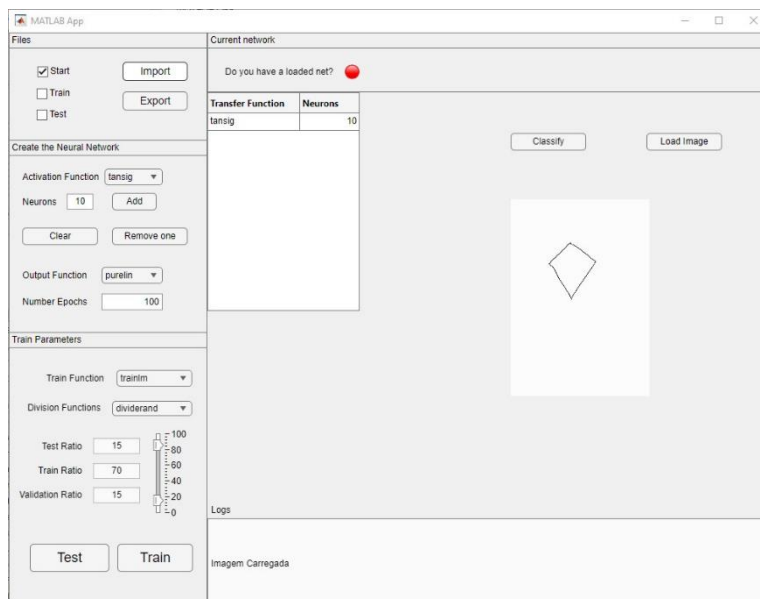


Figura 13: Aplicação gráfica – Carregamento e classificação de uma imagem

No âmbito da alínea e), foi desenvolvida uma aplicação gráfica em MATLAB com o objetivo de permitir ao utilizador interagir com redes neuronais de forma simples e intuitiva, cumprindo os requisitos propostos. A interface foi desenhada de modo a agrupar funcionalidades por áreas, promovendo uma navegação clara e lógica.

O painel de configuração da rede neuronal permite ao utilizador definir a topologia da rede (número de camadas e neurónios), selecionar as funções de ativação e treino, bem como ajustar o número de épocas e as percentagens de divisão entre treino, teste e validação. Estes parâmetros são facilmente manipuláveis através de menus suspensos e sliders, dispensando a necessidade de intervenção direta no código.

A aplicação inclui ainda funcionalidades para **treinar a rede**, **importar/exportar modelos já treinados**, e **carregar imagens** que podem ser classificadas com a rede selecionada. A estrutura modular da aplicação garante que cada uma destas ações é realizada de forma independente e intuitiva. O botão de classificação permite aplicar uma rede previamente treinada à imagem carregada, sendo a imagem exibida no painel gráfico. Existe também uma secção de “Logs” onde podem ser apresentadas mensagens informativas sobre o processo.

Em suma, a interface oferece uma solução coerente e funcional, permitindo ao utilizador configurar, treinar e aplicar redes neuronais sem necessidade de conhecimento técnico aprofundado, cumprindo de forma satisfatória os critérios estabelecidos.

6 CONCLUSÕES

O presente trabalho permitiu aplicar e aprofundar os conhecimentos adquiridos na unidade curricular de Conhecimento e Raciocínio, particularmente no desenvolvimento, treino e análise de redes neuronais feedforward para a classificação de formas geométricas. A abordagem seguiu uma estrutura progressiva, desde o treino com pequenos conjuntos de dados até à implementação de uma aplicação gráfica interativa.

Na **alínea a)**, foi possível atingir 100% de precisão com redes simples, confirmando que conjuntos de dados reduzidos não exigem arquiteturas complexas.

A **alínea b)** destacou a importância dos parâmetros da rede, principalmente a função de treino (com *trainbr* a obter os melhores resultados) e a função de ativação (*logsig* e *tansig*), enquanto evidenciou que redes mais profundas nem sempre trazem melhorias relevantes.

A **alínea c)** demonstrou que, embora algumas redes apresentem excelente desempenho nos dados de treino, nem todas generalizam bem para novos contextos, sendo a rede *best_net_1* a mais robusta.

A **alínea d)** evidenciou as limitações dos modelos face a dados fora do padrão (imagens desenhadas manualmente), onde as precisões caíram abaixo de 20%, sinalizando possíveis problemas de sobreajuste e a necessidade de treino com maior diversidade de dados.

Por fim, a **alínea e)** resultou numa aplicação gráfica funcional, intuitiva e adequada aos objetivos propostos, permitindo configurar, treinar e aplicar redes de forma autónoma e eficiente.

Concluindo, o projeto demonstrou não só a eficácia das redes neuronais em tarefas de classificação, como também os desafios associados à generalização e à variabilidade dos dados. A análise sistemática de múltiplos parâmetros permitiu identificar boas práticas de configuração e realçar a importância de um tratamento rigoroso dos dados. A aplicação final é um reflexo prático do conhecimento adquirido e constitui uma base sólida para trabalhos futuros nesta área.



**Instituto Superior
de Engenharia**

Politécnico de Coimbra