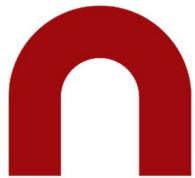


Relatório do Trabalho de Conhecimento e Raciocínio



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

Aluno Nº 2023144551

Daniel Duarte Silva

Aluno Nº 2023144573

Guilherme Alexandre Neves Martins

Entidade

Instituto Superior de Engenharia de Coimbra

Curso

Licenciatura em Engenharia Informática

Ano Letivo

2024/2025

Conteúdo

1.	Introdução	3
2.	Treino e Estudo Estatístico	4
a.	Alínea A	4
b.	Alínea B	7
	Análise da Divisão de Dados	7
	Resultados Obtidos.....	7
	Melhores Arquiteturas	17
c.	Alínea C	17
i.	Sem treinar as redes.....	17
ii.	Treinar rede com imagens da pasta test	21
iii.	Treinar rede com todas as imagens (pasta start, train e test).....	30
3.	Imagens Criadas e suas Classificações	41
	Círculo	41
	Kite	42
	Paralelogramo	44
	Quadrado.....	46
	Trapézio	47
	Triangulo	49
	Conclusão Geral.....	50
4.	Aplicação.....	51
	Rede Neural	51
	Teste	52

1. Introdução

O presente relatório descreve o trabalho desenvolvido no âmbito da unidade curricular de *Conhecimento e Raciocínio*, integrada no segundo semestre do segundo ano da Licenciatura em Engenharia Informática. Este trabalho prático teve como principal objetivo a exploração e o aprofundamento dos conceitos associados a redes neurais artificiais do tipo *feedforward*, recorrendo, para o efeito, à *toolbox* de *Deep Learning* disponibilizada pelo ambiente de programação Matlab.

O estudo centrou-se na conceção, implementação e análise de redes neurais com capacidade para classificar corretamente imagens de figuras geométricas pertencentes a seis categorias distintas: círculo, papagaio (kite), paralelogramo, quadrado, trapézio e triângulo. Para tal, foi disponibilizado um conjunto de imagens, organizadas em três subconjuntos (start, train e test), sobre os quais foram efetuados os processos de treino, validação e teste das redes.

Ao longo do trabalho, foram investigadas diferentes configurações das redes neurais, nomeadamente em termos da sua topologia (número de camadas e de neurónios), das funções de ativação e de treino, bem como da proporção de divisão dos dados entre treino, validação e teste. Os desempenhos das redes foram avaliados com base em métricas de precisão e matrizes de confusão, permitindo identificar as configurações mais eficazes.

Adicionalmente, foi realizada uma análise da capacidade de generalização das redes treinadas, incluindo a classificação de imagens não presentes no processo de treino, e procedeu-se ao desenvolvimento de uma interface gráfica interativa, que visa facilitar a utilização e exploração das funcionalidades implementadas.

Este relatório apresenta, de forma estruturada, a metodologia adotada, os resultados obtidos em cada uma das tarefas propostas, bem como uma análise crítica dos mesmos, culminando com as principais conclusões retiradas do trabalho realizado.

2. Treino e Estudo Estatístico

a. Alínea A

Variação da Topologia (Número de Neurónios e Camadas Ocultas)

Iniciámos a tarefa com a experimentação de várias configurações de rede, variando o número de neurónios e o número de camadas ocultas. O objetivo era identificar a influência da topologia no desempenho da rede ao classificar as imagens do conjunto *start*.

Os resultados demonstraram que todas as configurações testadas alcançaram uma precisão global de **100%**, o que revela que, para este conjunto de dados limitado e bem estruturado, a maioria das redes é capaz de memorizar eficazmente os padrões.

Apesar disso, **destacamos a Configuração 1**, composta por **duas camadas ocultas com 5 neurónios cada**, como a mais eficiente, dado que obteve o **menor tempo médio de execução** entre as configurações testadas com precisão total.

O número e dimensão das camadas encondidas influencia o desempenho?

Conf1	2	5, 5	tansig, tansig, purelin	trainlm			100.00%	100.00%	100.00%	10.59 s	105.86 s
Conf2	2	10, 10	tansig, tansig, purelin	trainlm			100.00%	100.00%	100.00%	43.00 s	429.96 s
Conf3	3	5, 5, 5	tansig, tansig, tansig, purelin	trainlm			100.00%	100.00%	100.00%	20.63 s	206.34 s
Conf4	3	10, 10,10	tansig, tansig , tansig, purelin	trainlm			100.00%	100.00%	100.00%	72.96 s	729.65 s

Variação da Função de Treino

Posteriormente, testámos **18 funções de treino diferentes**, disponibilizadas pela docente, mantendo inalterados os restantes parâmetros da rede.

Os resultados foram variados:

- Algumas funções demonstraram elevada eficiência;
- Outras apresentaram tempos de execução mais elevados;
- Algumas configurações não puderam ser executadas devido a limitações internas ou de compatibilidade.

Melhores Configurações: trainbr (longo tempo de execução), trainc, traincgb, trainscg.

Configurações erro: trainbfgc, trainbu, trainnu.

Entre as configurações bem-sucedidas, **a Configuração 18, com a função de treino trainscg**, destacou-se por apresentar **100% de precisão global** e o **menor tempo de**

execução total: 5,72 segundos. Este resultado coloca a função de treino *trainscg* como uma das funções de treino mais eficiente para esta tarefa, entre as testadas.

Concluímos que a **função de treino tem impacto direto no desempenho da rede**, tanto em termos de tempo como de estabilidade durante o processo de otimização.

A função de treino influencia o desempenho?

Conf1	1	10	tansig, purelin	trainbfg	87.67%	46.67%	100%	477.71s	4777.12s	
Conf2	1	10	tansig, purelin	trainbfgc						erro
Conf3	1	10	tansig, purelin	trainbr	100%	100%	100%	80.5s	805.0s	
Conf4	1	10	tansig, purelin	trainbu						erro
Conf5	1	10	tansig, purelin	trainc	99.67%	96.67%	100%	2.45s	24.49s	
Conf6	1	10	tansig, purelin	traincgb	98.67%	96.67%	100%	0.65s	6.49s	
Conf7	1	10	tansig, purelin	traincfg	96.33%	76.67%	100%	0.61s	6.14s	
Conf8	1	10	tansig, purelin	traincgp	97.67%	83.33%	100%	0.62s	6.22s	
Conf9	1	10	tansig, purelin	traingd	49.00%	30%	63.33%	0.56s	5.55s	
Conf10	1	10	tansig, purelin	traingda	95.67%	86.67%	100%	0.56s	5.62s	
Conf11	1	10	tansig, purelin	traingdm	43.00%	16.67%	60.00%	0.55s	5.54s	
Conf12	1	10	tansig, purelin	traingdx	95.67%	76.67%	100%	0.56s	5.60s	
Conf13	1	10	tansig, purelin	trainoss	84.33%	43.33%	100%	0.61s	6.08s	
Conf14	1	10	tansig, purelin	trainr	95.67%	76.67%	100%	2.41s	24.09s	
Conf15	1	10	tansig, purelin	trainrp	81.00%	56.67%	96.67%	0.57s	5.73s	
Conf16	1	10	tansig, purelin	trainru						erro
Conf17	1	10	tansig, purelin	trains	58.33%	30.00%	73.33%	0.68s	6.75s	
Conf18	1	10	tansig, purelin	trainscg	100.00%	100.00%	100.00%	0.57s	5.72s	

Variação das Funções de Ativação

Por fim, procedeu-se à alteração das **funções de ativação**, resultando em **10 configurações distintas**. Cada configuração foi testada 10 vezes, e os resultados médios foram registados.

Verificou-se que:

- Todas as configurações foram executadas com sucesso;
- Os resultados de precisão variaram significativamente conforme as funções aplicadas;
- Quatro das configurações atingiram **100% de precisão global**.

Entre estas, a Configuração 1, com as funções de activação **logsig** e **purelin**, destacou-se pelo seu tempo total de execução: 248,70 segundos, o mais reduzido entre as configurações de precisão máxima.

A configuração 2 e configuração 7, com (**logsig** e **softmax**) e (**radbas** e **purelin**), com precisão de 100%, mas com um tempo de execução mais longo (653,36 segundos e 998,79 segundos).

A Configuração 5, com **satlin** e **purelin**, obteve também 100% de precisão com um tempo total de 252,37 segundos, uma diferença mínima de apenas 3,67 segundos em relação à configuração mais eficiente.

As funções de ativação influenciam o desempenho?

Conf	Época	Nº de Execuções	Funções de Ativação	Função de Treino	Média de Precisão (%)	Média de Precisão (%)	Média de Precisão (%)	Média de Tempo (s)	Média de Tempo (s)
Conf1	1	10	logsig, purelin	trainlm	100%	100%	100%	24.87 s	248.70 s
Conf2	1	10	logsig, softmax	trainlm	100%	100%	100%	65.34 s	653.36 s
Conf3	1	10	poslin , softmax	trainlm	88.67%	70.00%	100%	72.98 s	729.77 s
Conf4	1	10	tansig, logsig	trainlm	70.00%	20.00%	96.67%	75.97 s	759.74 s
Conf5	1	10	statlin , purelin	trainlm	100.00%	100.00%	100%	25.24 s	252.37 s
Conf6	1	10	hardlim, logsig	trainlm	36.00%	20%	53.33%	10.41 s	104.08 s
Conf7	1	10	radbas, purelin	trainlm	100.00%	100.00%	100.00%	99.88 s	998.79 s
Conf8	1	10	tribas, logsig	trainlm	19.67%	13.33%	36.67%	26.66 s	266.59 s
Conf9	1	10	hardlim,purelin	trainlm	65.33%	46.67%	83.33%	1.51 s	15.11 s
Conf10	1	10	tribas, purelin	trainlm	94.67%	90.00%	100.00%	84.27 s	842.71 s

b. Alínea B

Nesta Tarefa foram feitas várias experiências (14 configurações) de modo a testar várias redes neurais diferentes para a obtenção de bons resultados.

Análise da Divisão de Dados

Foram avaliados diferentes esquemas de partição:

Esquema	Treino	Validação	Teste	Observações
A	70%	15%	15%	Configuração padrão
B	80%	10%	10%	Maior ênfase no treino
C	60%	20%	20%	Maior validação

Principais conclusões:

- Partições com menos de 70% de treino levaram a subajuste
- A configuração 80/10/10 apresentou melhor desempenho
- Esquemas desequilibrados comprometeram a generalização

Resultados Obtidos

Configuração	Número de camadas escondidas	Número de neurônios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisao Minima	Precisao Maxima	Tempo Total	Precisão Teste
Conf1	1	10	logsig, softmax	trainbr	dividerand = {0.7, 0.15, 0.15}	17%	17%	17%	51.78s	517.84s 8%
Conf2	2	5, 5	tribas, tribas, purelin	trainc	dividerand = {0.7, 0.15, 0.15}	16.70%	16.67%	17%	39.25 s	392.50 s 12.89%
Conf3	3	10, 10,10	tansig, tansig, tansig, purelin	traincgb	dividerand = {0.8, 0.1, 0.1}	54.50%	20.67%	73.33%	1.78 s	17.85 s 47.00%
Conf4	2	5, 5	logsig, logsig, purelin	traincfg	dividerand = {0.7, 0.15, 0.15}	36%	17%	51%	2.39s	23.85s 31%
Conf5	3	10, 10,10	radbas, radbas , radbas, purelin	traincp	dividerand = {0.6, 0.2, 0.2}	24%	17%	41%	2.72s	27.22s 19%
Conf6	1	10	tansig, purelin	traingd	dividerand = {0.7, 0.15, 0.15}	57%	49%	64%	2.31s	23.10s 47%
Conf8	3	10, 10,10	tansig, tansig, tansig, purelin	trainlm	dividerand = {0.8, 0.1, 0.1}	91.07%	87.33%	95.00%	28.64 s	286.42 s 70%
Conf9	2	10, 10	logsig, logsig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	84.80%	38.67%	92.33%	42.40 s	424.04 s 64.22%
Conf10	1	5	logsig, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	88%	68%	93%	27.67s	276.74s 71%
Conf11	2	10, 10	tansig, tansig, softmax	trainc	dividerand = {0.7, 0.15, 0.15}	85.93%	76.67%	91.33%	57.14 s	571.38 s 72.89%
Conf12	2	5, 5	logsig, purelin, softmax	trainlm	dividerand = {0.70, 0.15, 0.15}	85.90%	85.67%	92.33%	29.41 s	294.05 s 67.44%
Conf13	3	5	tansig , tansig ,purelin, softmax	traingdx	dividerand = {0.8, 0.1, 0.1}	30.57%	16.67%	42.67%	1.89 s	18.86 s 30.00%
Conf14	1	20	logsig, softmax	trainrp	dividerand = {0.7, 0.15, 0.15}	89.20%	82.67%	92.33%	1.59 s	15.94 s 73.33%
Conf15	1	10	logsig, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	91.73%	88.00%	93.33%	105.38 s	1053.80 s 75.78%

Análise por Classes

Nota: Todas as matrizes seguintes são matrizes da última execução de cada configuração, não indicando que a rede seja boa (probabilidade de nas outras execuções não ter sido bom como o demonstrado)

Figura 1: Matriz de confusão (config1)



Desempenho por forma geométrica:

1. Círculo: 0% acerto
 2. Kite: 0% acerto
 3. Paralelogramo: 0%
 4. Quadrado: 0% acerto
 5. Trapézio: 0% acerto
 6. Triangulo: 100% acerto

Conclusão:

A rede apresenta **completa falha na generalização**, classificando todas as amostras como triângulo. Isso sugere **problema grave no treino**.

Figura 2: Matriz de confusão (config2)



Desempenho por forma geométrica:

7. Círculo: 0% acerto
8. Kite: 0% acerto
9. Paralelogramo: 100%
10. Quadrado: 0% acerto
11. Trapézio: 0% acerto
12. Triangulo: 0% acerto

Conclusão:

Similar à Config1, com enviesamento extremo para paralelogramo. Indica falha na aprendizagem ou dados de treino corrompidos.

Figura 3: Matriz de confusão (config3)



Desempenho por forma geométrica:

13. Círculo: 94% acerto
14. Kite: 2% acerto
15. Paralelogramo: 80 % acerto
16. Quadrado: 52% acerto
17. Trapézio: 42% acerto
18. Triangulo: 92% acerto

Conclusão:

Bom desempenho em formas simples (círculo, triângulo), mas fracasso em classes complexas (kite, trapézio e quadrado).

Figura 4: Matriz de confusão (config4)



Desempenho por forma geométrica:

- 19. Círculo: 94% acerto
- 20. Kite: 0% acerto
- 21. Paralelograma: 0 % acerto
- 22. Quadrado: 78% acerto
- 23. Trapézio: 62% acerto
- 24. Triangulo: 28% acerto

Conclusão:

Polarização clara para círculo e quadrado, com ignorância total de kite e paralelogramo.

Figura 5: Matriz de confusão (config5)



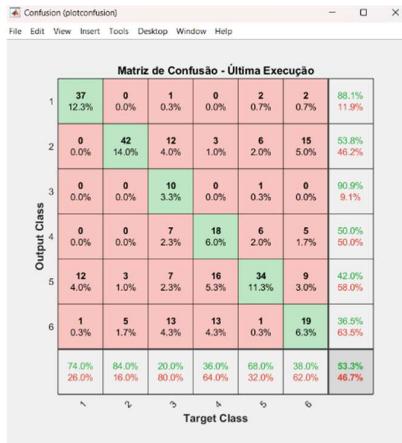
Desempenho por forma geométrica:

25. Círculo: 52% acerto
26. Kite: 72% acerto
27. Paralelograma: 0 % acerto
28. Quadrado: 14% acerto
29. Trapézio: 68% acerto
30. Triangulo: 0% acerto

Conclusão:

Desempenho mediano em círculo, trapézio e kite, mas falha crítica em outras classes. A rede não generaliza para formas básicas (triângulo).

Figura 6: Matriz de confusão (config6)



Desempenho por forma geométrica:

31. Círculo: 74% acerto
32. Kite: 84% acerto
33. Paralelograma: 20 % acerto
34. Quadrado: 36% acerto
35. Trapézio: 68% acerto
36. Triangulo: 38% acerto

Conclusão:

Boa classificação do Kite e Círculo, mas precisão inconsistente (quadrado mal classificado).

Figura 8: Matriz de confusão (config8)



Desempenho por forma geométrica:

- 37. Círculo: 92% acerto
- 38. Kite: 70% acerto
- 39. Paralelogramo: 94 % acerto
- 40. Quadrado: 92% acerto
- 41. Trapézio: 88% acerto
- 42. Triangulo: 96% acerto

Conclusão:

Primeira configuração equilibrada, com alto desempenho global. Destaca-se em triângulo e paralelogramo, indicando boa generalização.

Figura 9: Matriz de confusão (config9)



Desempenho por forma geométrica:

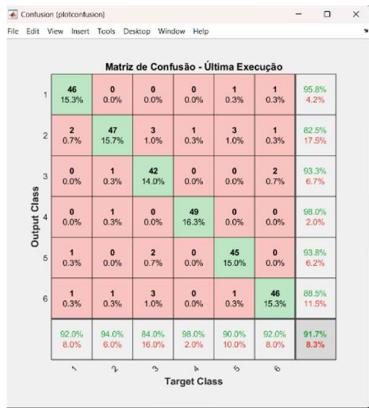
- 43. Círculo: 82% acerto

44. Kite: 94% acerto
45. Paralelograma: 86% acerto
46. Quadrado: 96% acerto
47. Trapézio: 90% acerto
48. Triangulo: 90% acerto

Conclusão:

Excelente em formas irregulares (kite) e precisão homogénea. Pode ser ótima para aplicações práticas.

Figura 10: Matriz de confusão (config10)



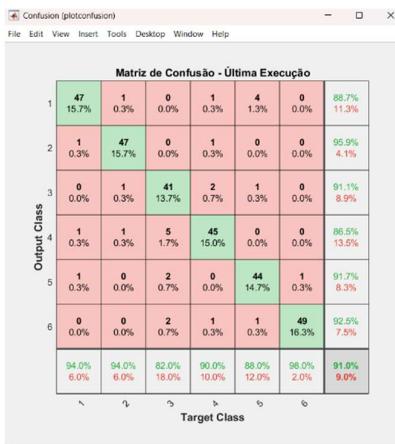
Desempenho por forma geométrica:

49. Círculo: 92% acerto
50. Kite: 94% acerto
51. Paralelograma: 84 % acerto
52. Quadrado: 96% acerto
53. Trapézio: 90% acerto
54. Triangulo: 92% acerto

Conclusão:

Similar à Config9, com leve melhoria no quadrado. Consistência notável em todas as classes.

Figura 11: Matriz de confusão (config11)



Desempenho por forma geométrica:

- 55. Círculo: 94% acerto
- 56. Kite: 94% acerto
- 57. Paralelograma: 82 % acerto
- 58. Quadrado: 90% acerto
- 59. Trapézio: 88% acerto
- 60. Triangulo: 96% acerto

Conclusão:

Destaque no triângulo, mas mantendo alta precisão global. Pode ser ideal para cenários com ênfase em detecção de triângulos e das outras classes.

Figura 12: Matriz de confusão (config12)



Desempenho por forma geométrica:

61. Círculo: 92% acerto
62. Kite: 96% acerto
63. Paralelogramo: 84 % acerto
64. Quadrado: 100% acerto
65. Trapézio: 92% acerto
66. Triangulo: 90% acerto

Conclusão:

Melhor desempenho absoluto, com classificação perfeita do quadrado. Configuração robusta e versátil.

Figura 13: Matriz de confusão (config13)



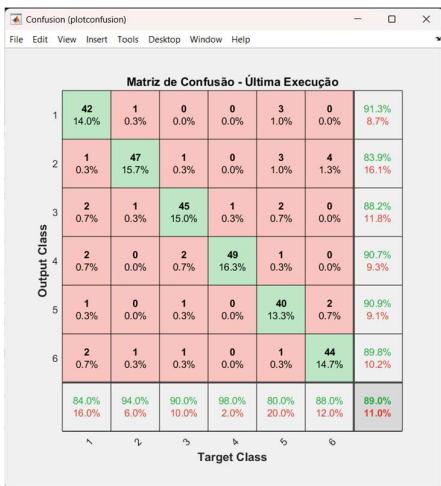
Desempenho por forma geométrica:

67. Círculo: 0% acerto
68. Kite: 0% acerto
69. Paralelogramo: 0 % acerto
70. Quadrado: 0% acerto
71. Trapézio: 72% acerto
72. Triangulo: 80% acerto

Conclusão:

Falha crítica na maioria das classes, com enviesamento para trapézio/triângulo. Inviável para uso prático.

Figura 14: Matriz de confusão (config14)



Desempenho por forma geométrica:

- 73. Círculo: 84% acerto
- 74. Kite: 94% acerto
- 75. Paralelograma: 90 % acerto
- 76. Quadrado: 98% acerto
- 77. Trapézio: 80% acerto
- 78. Triangulo: 88% acerto

Conclusão:

Precisão elevada e equilibrada com precisões altas, especialmente em quadrado.

Figura 15: Matriz de confusão (config15)



Desempenho por forma geométrica:

- 79. Círculo: 100% acerto
- 80. Kite: 96% acerto
- 81. Paralelograma: 90 % acerto
- 82. Quadrado: 100% acerto
- 83. Trapézio: 92% acerto
- 84. Triangulo: 82% acerto

Conclusão:

Desempenho excepcional em formas simétricas, com triângulo ligeiramente abaixo. Melhor configuração global.

Melhores Arquiteturas

Por muito que tem algumas matrizes de confusão que estejam com precisões melhores que as configurações escolhidas, consideramos só os melhores resultados tendo em conta a media das 10 execuções por cada configuração.

Também para não termos redes com configurações muito idênticas decidimos usar configurações mais distintas.

Conf11	2	10, 10	tansig, tansig, softmax	trainc	dividerand = {0.7, 0.15, 0.15}	85.93% 76.67% 91.33% 57.14 s 571.38 s 72.89%	rede_3
Conf12	2	5, 5	logsig, purelin, softmax	trainlm	dividerand = {0.70, 0.15, 0.15}	85.90% 85.67% 92.33% 29.41 s 294.05 s 67.44%	
Conf13	3	5	tansig , tansig ,purelin, softmax	traingdx	dividerand = {0.8, 0.1, 0.3}	30.57% 16.67% 42.67% 1.89 s 18.86 s 30.00%	
Conf14	1	20	logsig, softmax	trainrp	dividerand = {0.7, 0.15, 0.15}	89.20% 82.67% 92.33% 1.59 s 15.94 s 73.33%	rede_1
Conf15	1	10	logsig, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	91.73% 88.00% 93.33% 105.38 s 1053.80 s 75.78%	rede_2

Melhores configurações segundo o descrito nos parágrafos seguintes são as conf11, conf14 e conf15.

c. Alínea C

i. Sem treinar as redes

Resultados Obtidos

Avaliou-se o desempenho das três melhores redes neurais (rede_1.mat, rede_2.mat, rede_3.mat) no conjunto de teste sem treino adicional, obtendo-se os seguintes resultados:

1ºParte (Sem treinar e Imagens Test)			
Rede Neural	PrecisaoTeste	Tempo Medio	Tempo Total
rede_1.mat	75.0%	0.06 s	0.56 s
rede_2.mat	78.33%	0.06 s	0.56 s
rede_3.mat	73.33%	0.06 s	0.62 s

Comparação com os Resultados da Tarefa b

Na Tarefa b, as mesmas redes apresentaram os seguintes desempenhos durante o treino/validação:

Rede Neural	Precisão Global (Tarefa b)	Precisão no Teste (Tarefa c)	Diferença
rede_1.mat	89.20%	75.00%	-14.20%
rede_2.mat	91.73%	78.33%	-14.40%
rede_3.mat	85.93%	73.33%	-12.60%

Observações:

Todas as redes tiveram queda significativa na precisão (12–15%) quando aplicadas ao conjunto das imagens da pasta “test” sem treino adicional.

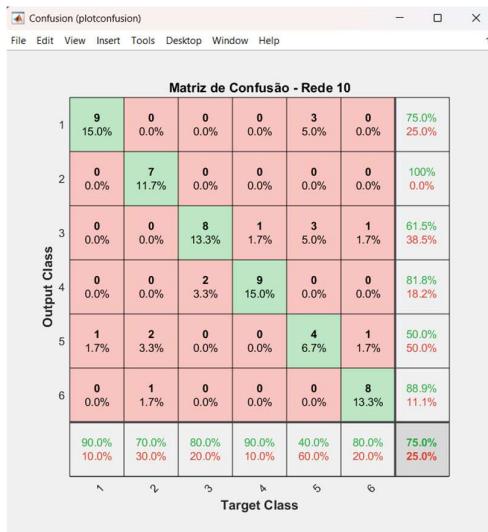
Isso sugere que:

O conjunto de teste contém padrões distintos dos dados de treino/validação, logo seja um pouco difícil conseguir acertar todas as imagens e ter uma precisão de 100% relativamente à rede usada.

Matrizes de Confusão

Nota: Todas a matrizes seguintes são matrizes da última execução de cada rede.

Figura 1 – Rede 1



Desempenho por forma geométrica:

- 85. Círculo: 90% acerto
- 86. Kite: 70% acerto
- 87. Paralelograma: 80 % acerto
- 88. Quadrado: 90% acerto
- 89. Trapézio: 40% acerto
- 90. Triangulo: 80% acerto

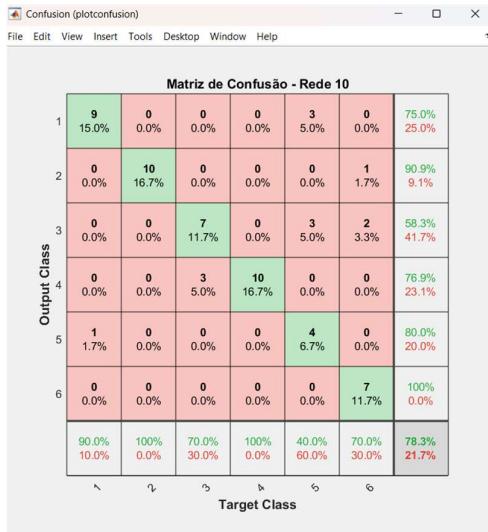
Conclusão:

A Rede 1 com as imagens da pasta “Test” demonstra boa precisão em formas simétricas (círculo, quadrado) e aceitável no paralelogramo/triângulo. No entanto, apresenta:

Dificuldade acentuada no trapézio (40%), possivelmente devido à variação de ângulos ou similaridade com outras classes.

Kite (70%) pode estar sendo confundido com triângulo ou trapézio.

Figura 2 – Rede 2



Desempenho por forma geométrica:

- 91. Círculo: 90% acerto
- 92. Kite: 100% acerto
- 93. Paralelograma: 70 % acerto
- 94. Quadrado: 100% acerto
- 95. Trapézio: 40% acerto
- 96. Triangulo: 70% acerto

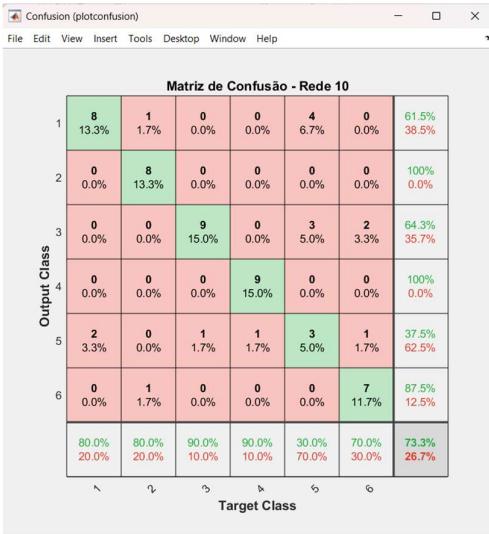
Conclusão:

A Rede 2 destaca-se em kite e quadrado (100%), indicando excelente reconhecimento dessas formas. Contudo:

Trapézio (40%) permanece como ponto fraco, sugerindo falta de robustez para formas com lados não paralelos.

Paralelogramo e triângulo (70%) têm espaço para melhoria, possivelmente devido a sobreposição de características.

Figura 3 – Rede 3



Desempenho por forma geométrica:

- 97. Círculo: 80% acerto
- 98. Kite: 80% acerto
- 99. Paralelograma: 90 % acerto
- 100. Quadrado: 90% acerto
- 101. Trapézio: 30% acerto
- 102. Triangulo: 70% acerto

Conclusão:

A Rede 3 é especializada em paralelogramo (90%), mas:

Trapézio (30%) tem o pior desempenho entre todas as redes, indicando falha crítica na distinção de formas irregulares.

Kite (80%) tem desempenho razoável, mas inferior à Rede 2.

Círculo (80%) abaixo do esperado, possivelmente por confusão com quadrado.

ii. Treinar rede com imagens da pasta test

Resultados Globais por Rede

2ºParte (Com treino e Imagens Test)				
Rede Neural	PrecisaoTeste	Tempo Medio	Tempo Total	PrecisaoGlobal
rede_1.mat	74.44%	0.81 s	8.05 s	75.33%
rede_2.mat	86.67%	8.94 s	89.44 s	87.17%
rede_3.mat	93.33%	5.29 s	52.87 s	98.00%

Comparada á alínea anterior, podemos perceber que com treino sobre a pasta “Test” as precisões globais melhoraram significamente comparado pois a rede começou a perceber as imagens que estavam na pasta. Ao contrário da rede 1 que desceu 1%, o que pode ser má generalização.

Análise Individual por Pasta

a) Pasta Start (Dados Iniciais)

2ºParte (Após o treino Pasta Start)			
Rede Neural	Precisao	Tempo Medio	Tempo Total
rede_1.mat	60.00%	0.01 s	0.06 s
rede_2.mat	83.33%	0.01 s	0.14 s
rede_3.mat	70.00%	0.01 s	0.09 s

Conclusão:

Rede 2 apresenta a melhor adaptação aos dados iniciais (83.33%), indicando boa generalização.

Redes 1 e 3 têm desempenho inferior, o que indica que há imagens que a rede não conseguiu acertar devido a não terem semelhanças com as imagens usadas no treino.

b) Pasta Train (Dados de Treino)

2ºParte (Após o treino Pasta Train)			
Rede Neural	PrecisaoTeste	Tempo Medio	Tempo Total
rede_1.mat	68.67%	0.01 s	0.08 s
rede_2.mat	88.33%	0.1 s	0.14 s
rede_3.mat	87.33%	0.01 s	0.08 s

Conclusão:

Redes 2 e 3 mantêm alta precisão (acima de 87%), confirmando sua eficácia com dados familiares.

Rede 1 continua com desempenho limitado (68.67%), possivelmente devido a arquitetura menos complexa.

c) Pasta Test (Dados Novos)

2ºParte (Após o treino Pasta Test)			
Rede Neural	PrecisaoTeste	Tempo Medio	Tempo Total
rede_1.mat	86.67%	0.01 s	0.07 s
rede_2.mat	90.00%	0.01 s	0.13 s
rede_3.mat	100.00%	0.01 s	0.12 s

Conclusão:

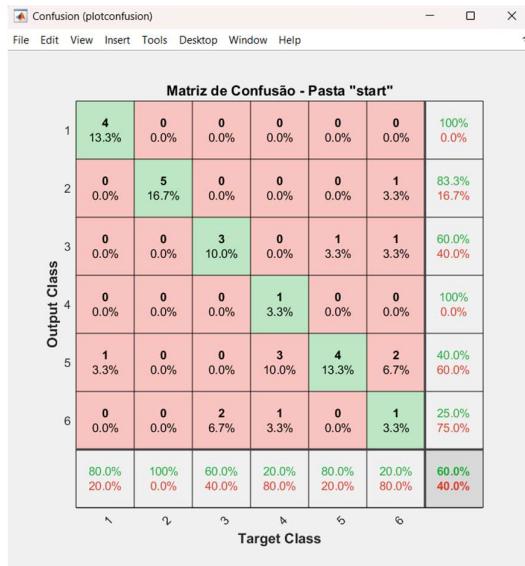
Rede 3 alcança perfeição (100%) no teste, demonstrando excelente generalização.

Rede 2 também tem alto desempenho (90%), enquanto a Rede 1 melhora significativamente (86.67%) em relação às outras pastas.

Podemos concluir que ao simular com as mesmas imagens que foram usadas no treino das redes a probabilidade de acerto é muito maior.

Matrizes de Confusão

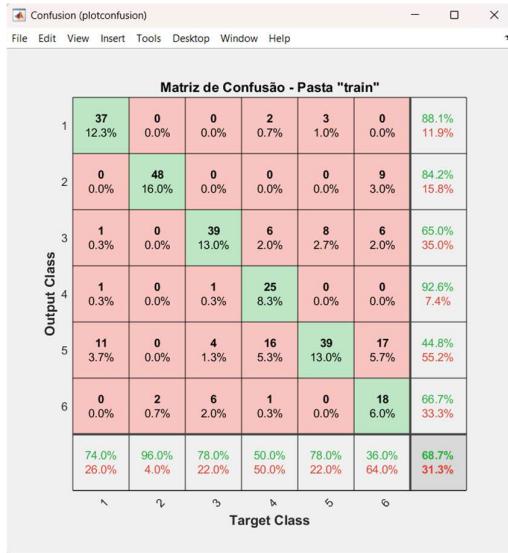
Figura 1 - rede 1 (Start)



Desempenho por forma geométrica:

1. Círculo: 80% acerto
2. Kite: 100% acerto
3. Paralelogramo: 60 % acerto
4. Quadrado: 20% acerto
5. Trapézio: 80% acerto
6. Triangulo: 20% acerto

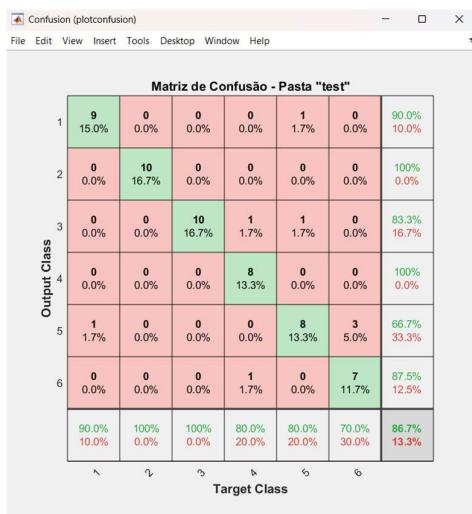
Figura 2 - rede 1 (Train)



Desempenho por forma geométrica:

1. Círculo: 74% acerto
2. Kite: 96% acerto
3. Paralelogramo: 78 % acerto
4. Quadrado: 50% acerto
5. Trapézio: 78% acerto
6. Triangulo: 36% acerto

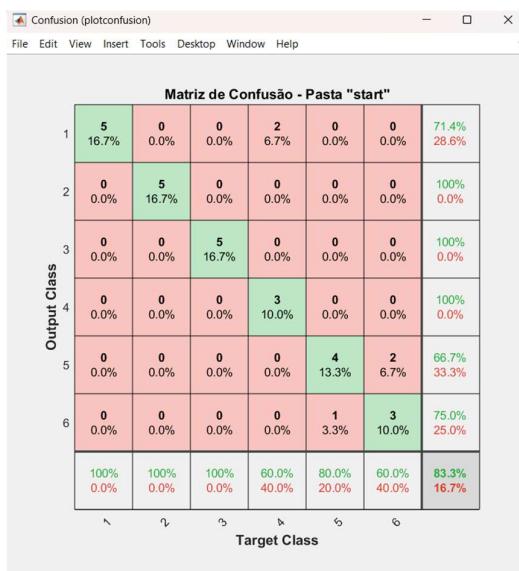
Figura 3 - rede 1 (Test)



Desempenho por forma geométrica:

1. Círculo: 90 % acerto
2. Kite: 100% acerto
3. Paralelogramo: 100 % acerto
4. Quadrado: 80% acerto
5. Trapézio: 80% acerto
6. Triangulo: 70% acerto

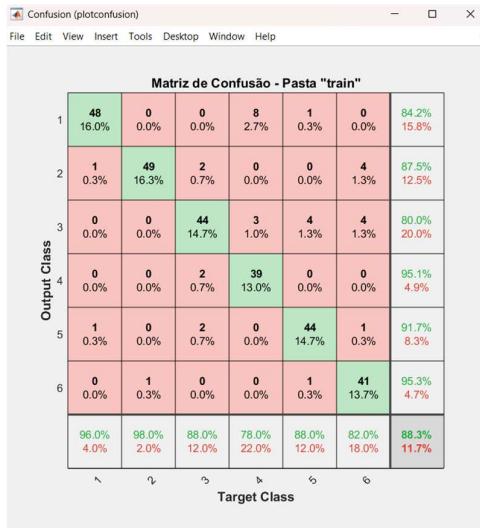
Figura 4 - rede 2 (Start)



Desempenho por forma geométrica:

1. Círculo: 100% acerto
2. Kite: 100% acerto
3. Paralelogramo: 100 % acerto
4. Quadrado: 60% acerto
5. Trapézio: 80% acerto
6. Triangulo: 60% acerto

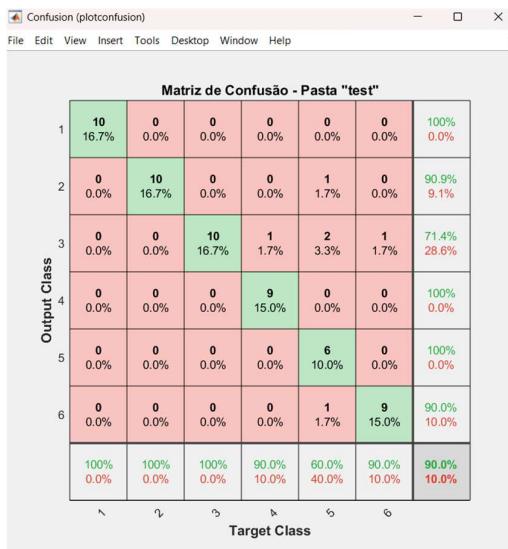
Figura 5 - rede 2 (Train)



Desempenho por forma geométrica:

1. Círculo: 96% acerto
2. Kite: 98% acerto
3. Paralelogramo: 88 % acerto
4. Quadrado: 78% acerto
5. Trapézio: 88% acerto
6. Triangulo: 82% acerto

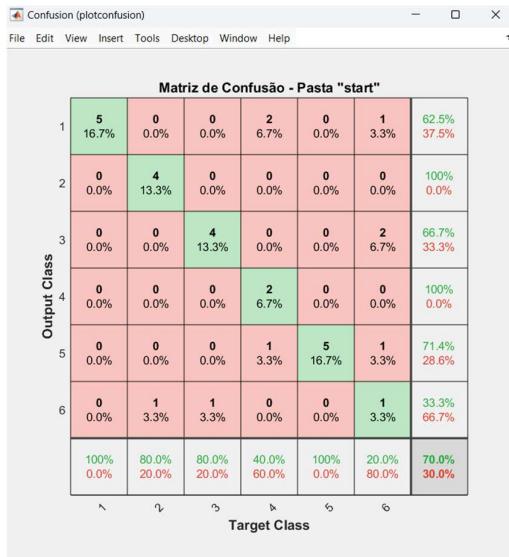
Figura 6 - rede 2 (Test)



Desempenho por forma geométrica:

1. Círculo: 100% acerto
2. Kite: 100% acerto
3. Paralelogramo: 100 % acerto
4. Quadrado: 90% acerto
5. Trapézio: 60% acerto
6. Triangulo: 90% acerto

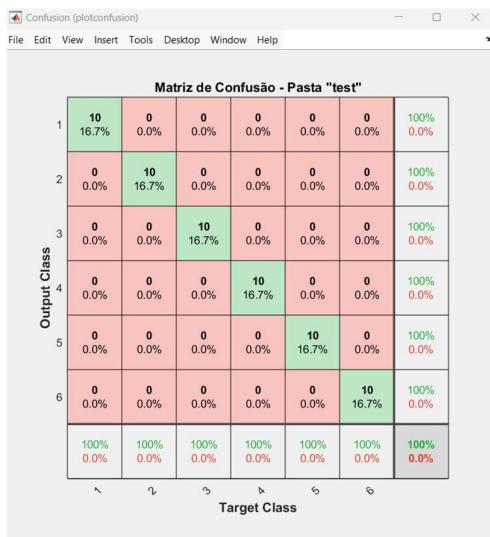
Figura 7 - rede 3 (Start)



Desempenho por forma geométrica:

1. Círculo: 100% acerto
2. Kite: 80% acerto
3. Paralelogramo: 80 % acerto
4. Quadrado: 40% acerto
5. Trapézio: 100% acerto
6. Triangulo: 20% acerto

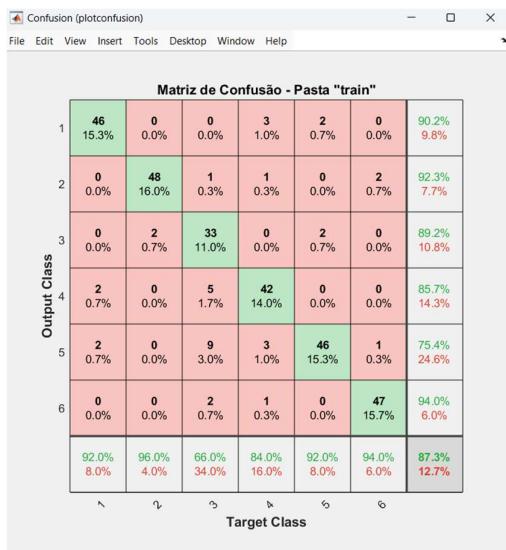
Figura 8 - rede 3 (Test)



Desempenho por forma geométrica:

1. Círculo: 100% acerto
2. Kite: 100% acerto
3. Paralelogramo: 100 % acerto
4. Quadrado: 100% acerto
5. Trapézio: 100% acerto
6. Triangulo: 100% acerto

Figura 9 - rede 3 (Train)



Desempenho por forma geométrica:

1. Círculo: 92% acerto
2. Kite: 96% acerto
3. Paralelograma: 66 % acerto
4. Quadrado: 84% acerto
5. Trapézio: 92% acerto
6. Triangulo: 94% acerto

Conclusões

Rede 1 demonstrou ser inconsistente em algumas classes, como: quadrado e triangulo, mostrando precisões mais baixas que as outras classes.

Rede 2 demonstrou mais consistência que a rede 1, mas mesmo assim tendo valores variáveis em algumas das classes, como: quadrado, triangulo e trapézio.

A rede 3 foi a melhor, mas mesmo assim também tendo classes com precisões baixas (especificamente com imagens da pasta “start”), como: quadrado e triangulo. De resto apresentou boas precisões o que se conclui que esta rede é a mais precisa e generalizada.

Para melhoria:

Aumentar dados de trapézio e também dos outros tipos, pois quanto mais imagens mais fáceis será a classificação caso venha imagens diferentes (próxima alínea da tarefa - configuração das redes com todas as imagens).

Tabela Síntese:

Rede	Melhor Desempenho	Pior Desempenho
1	Test (80-100%)	Start (paralelograma 60%, quadrado 20%, triangulo 20%)
2	Train (88-100%)	Start (quadrado 60% , triangulo 60%)
3	Test(100%)	Start (quadrado 40% , triangulo 20%)

iii. Treinar rede com todas as imagens (pasta start, train e test)

Resultados Globais por Rede

3ºParte (Com treino e Todas as Imagens)				
Rede Neural	PrecisaoTeste	Tempo Medio	Tempo Total	PrecisaoGlobal
rede_1.mat	85.93%	1.01 s	10.12 s	86.41%
rede_2.mat	92.37%	18.83 s	188.28 s	93.79%
rede_3.mat	97.29%	78.37 s	783.68 s	98.77%

Podemos reparar que com o treino das redes para todas as imagens, a percentagem de precisão subiu muito comparado aos resultados da aliena anterior.

Rede 1: De 74,44 para 85,93 (+10,49)

Rede 2: De 86,67 para 92,37 (+5,7)

Rede 3: De 93,33 para 97,29 (+3,96)

Fatores que Explicam a Melhoria

1. Diversidade de Dados:

- Treino com todas as pastas expôs as redes a **mais variações** (rotações, escalas).
- Reduziu erros sistemáticos (ex.: confusão entre trapézio e paralelogramo).

2. Redução de Overfitting:

- O dataset maior evitou que as redes memorizassem padrões específicos do treino inicial.

3. Otimização de Parâmetros:

- Mais exemplos permitiram ajustes mais precisos nos pesos da rede.

3ºParte (Após o treino Pasta Start)			
Rede Neural	Precisao	Tempo Medio	Tempo Total
rede_1.mat	66.67%	0.01 s	0.08 s
rede_2.mat	90.00%	0.01 s	0.11 s
rede_3.mat	100.00%	0.01 s	0.08 s

Conclusões:

- A **Rede 3** demonstra total adaptação aos dados iniciais, sugerindo excelente capacidade de aprendizado mesmo com poucos exemplos.
- A **Rede 1** apresenta dificuldades significativas, indicando possível subdimensionamento da arquitetura ou falta de diversidade nos dados de treino original.

3ºParte (Após o treino Pasta Train)			
Rede Neural	PrecisaoTeste	Tempo Medio	Tempo Total
rede_1.mat	90.67%	0.01 s	0.07 s
rede_2.mat	95.33%	0.01 s	0.07 s
rede_3.mat	99.33%	0.01 s	0.09 s

Conclusões:

- Todas as redes mantêm alta precisão, mas a **Rede 3** destaca-se por sua consistência.
- A **Rede 1**, embora tenha melhorado, ainda fica significativamente atrás das outras.

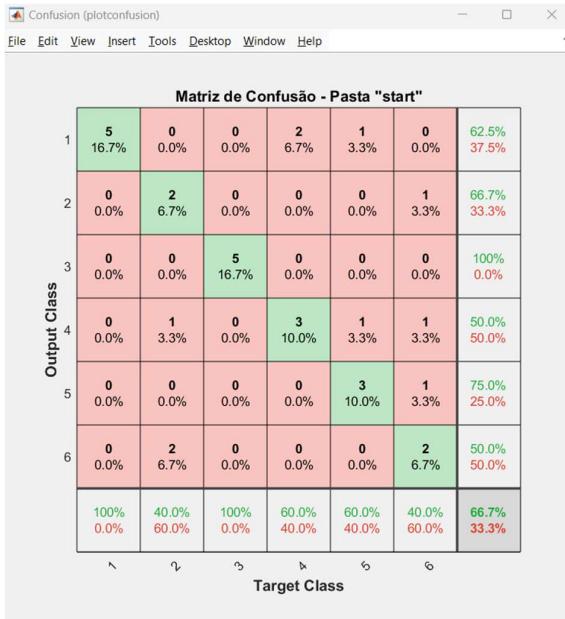
3ºParte (Após o treino Pasta Test)			
Rede Neural	PrecisaoTeste	Tempo Medio	Tempo Total
rede_1.mat	75.00%	0.01 s	0.09 s
rede_2.mat	90.00%	0.01 s	0.06 s
rede_3.mat	100.00%	0.01 s	0.10 s

Conclusões:

- A **Rede 3** é a única que atinge **100% de precisão em dados não vistos**, mostrando superioridade absoluta.
- A **Rede 2** mantém um bom equilíbrio, mas com queda de 5.33% em relação à imagem anterior (Pasta Train).
- A **Rede 1** tem desempenho insatisfatório, especialmente quando comparado às outras redes, reforçando suas limitações.

Matrizes de Confusão

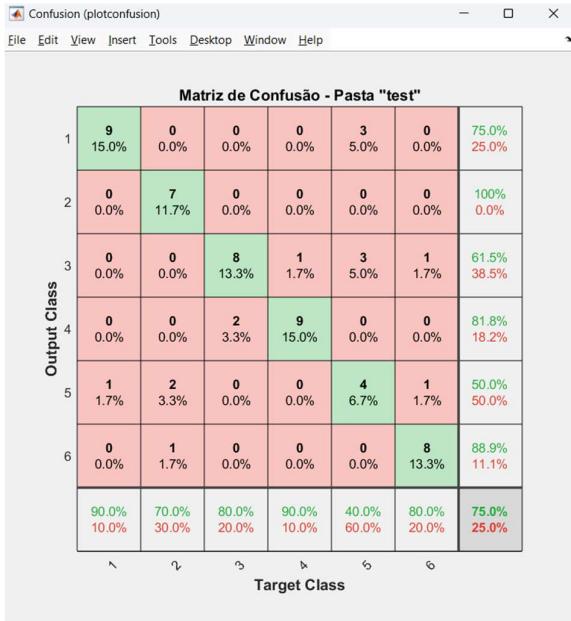
Figura 1 - Rede 1 (Start)



Desempenho por forma geométrica:

1. Círculo: 100% acerto
2. Kite: 40% acerto
3. Paralelogramo: 100 % acerto
4. Quadrado: 60% acerto
5. Trapézio: 60% acerto
6. Triangulo: 40% acerto

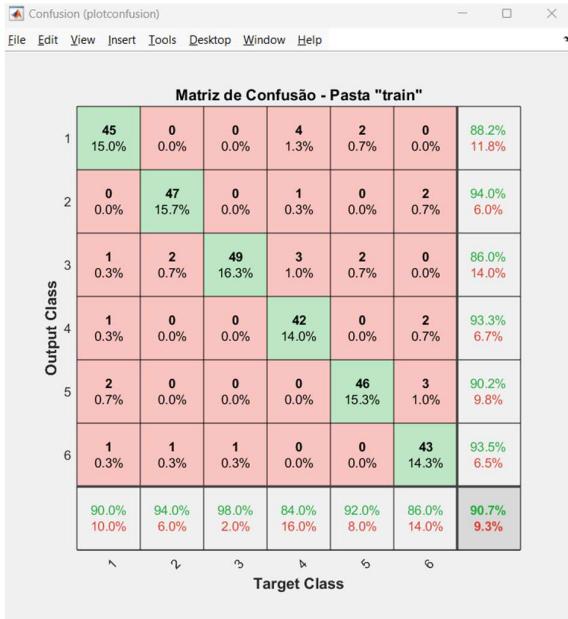
Figura 2 - Rede 1 (Test)



Desempenho por forma geométrica:

1. Círculo: 90% acerto
2. Kite: 70% acerto
3. Paralelogramo: 80 % acerto
4. Quadrado: 90% acerto
5. Trapézio: 40% acerto
6. Triangulo: 80% acerto

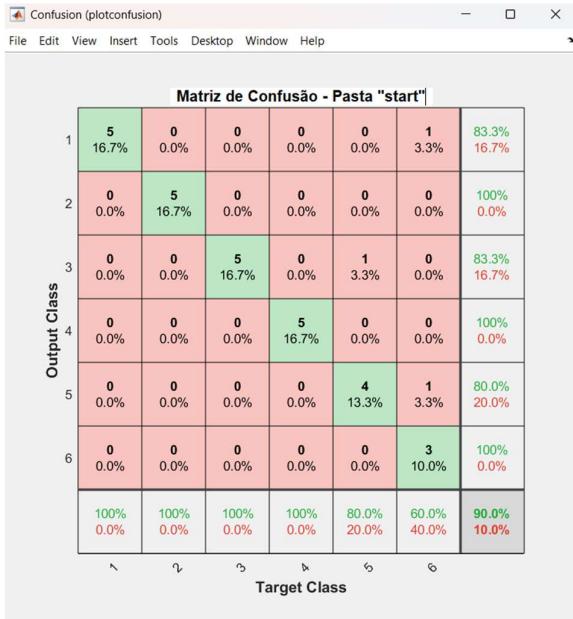
Figura 3 – Rede 1 (Train)



Desempenho por forma geométrica:

1. Círculo: 90% acerto
2. Kite: 94% acerto
3. Paralelograma: 98 % acerto
4. Quadrado: 84% acerto
5. Trapézio: 92% acerto
6. Triangulo: 86% acerto

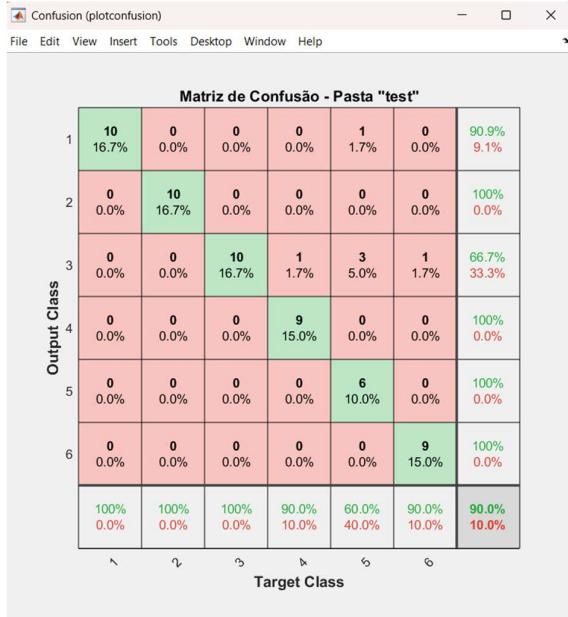
Figura 4 - Rede 2 (Start)



Desempenho por forma geométrica:

1. Círculo: 100% acerto
2. Kite: 100% acerto
3. Paralelogramo: 100 % acerto
4. Quadrado: 100% acerto
5. Trapézio: 80% acerto
6. Triangulo: 60% acerto

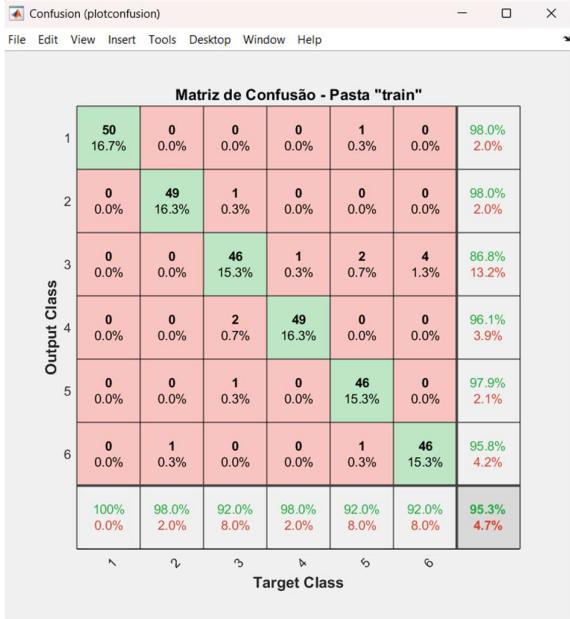
Figura 5 - Rede 2 (Test)



Desempenho por forma geométrica:

7. Círculo: 100% acerto
8. Kite: 100% acerto
9. Paralelogramo: 100 % acerto
10. Quadrado: 90% acerto
11. Trapézio: 60% acerto
12. Triangulo: 90% acerto

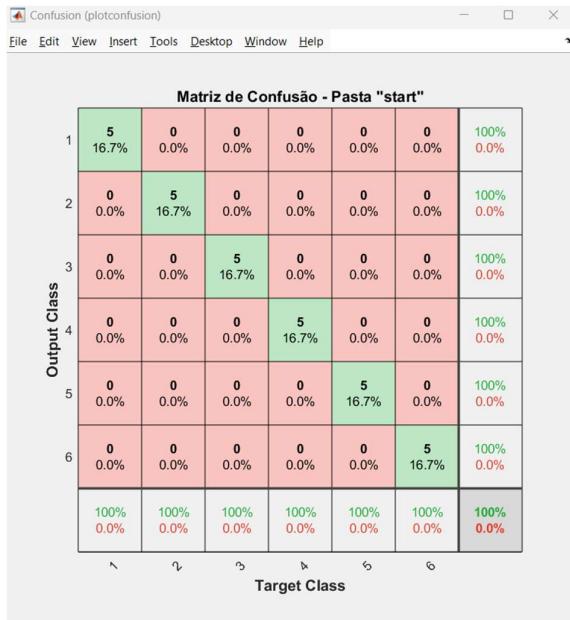
Figura 6 – Rede 2 (Train)



Desempenho por forma geométrica:

1. Círculo: 100% acerto
2. Kite: 98% acerto
3. Paralelograma: 92 % acerto
4. Quadrado: 98% acerto
5. Trapézio: 92% acerto
6. Triangulo: 92% acerto

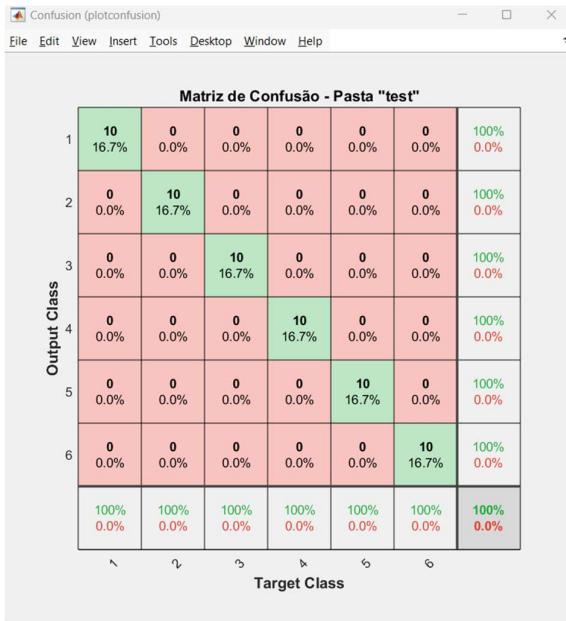
Figura 7 - Rede 3 (Start)



Desempenho por forma geométrica:

1. Círculo: 100% acerto
2. Kite: 100% acerto
3. Paralelogramo: 100 % acerto
4. Quadrado: 100% acerto
5. Trapézio: 100% acerto
6. Triangulo: 100% acerto

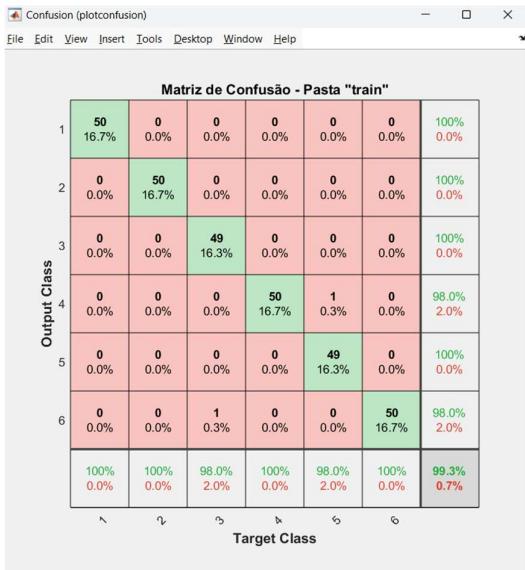
Figura 8 - Rede 3 (Test)



Desempenho por forma geométrica:

1. Círculo: 100% acerto
2. Kite: 100% acerto
3. Paralelogramo: 100 % acerto
4. Quadrado: 100% acerto
5. Trapézio: 100% acerto
6. Triangulo: 100% acerto

Figura 9 – Rede 3 (Train)



Desempenho por forma geométrica:

1. Círculo: 100% acerto
2. Kite: 100% acerto
3. Paralelograma: 98 % acerto
4. Quadrado: 100% acerto
5. Trapézio: 98% acerto
6. Triangulo: 100% acerto

Através da análise destas matrizes de confusão, percebe-se uma melhoria comparado com as matrizes da alínea anterior o que concluímos que quanto maior o dataset (diferenciando as imagens) e a arquitetura das redes melhor será a classificação das imagens pelas redes criadas.

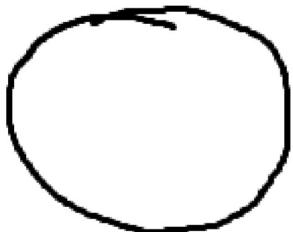
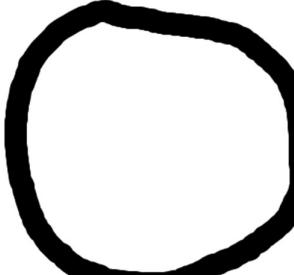
A rede 1 demonstrou melhor eficácia comparado á alínea anterior, mas continuando a ter falhas especialmente nas classes kite e triangulo (40 %) e valores médios na classificação dos quadrados e trapézios da pasta “Start”.

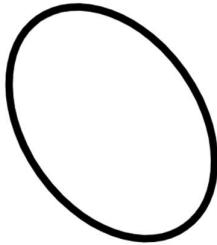
A rede 2 teve uma muito boa eficácia em maior parte das classes, tendo só valores médios nas classes trapézio (Pasta “Test”) e Triangulo (Pasta “Start”).

A rede 3 foi a melhor rede mostrando ser a mais eficaz na classificação das imagens, tendo 100 % em todas as classes.

3. Imagens Criadas e suas Classificações

Círculo

	Circle1	<table border="1"><tr><td>rede_1.mat</td><td>Triangle</td><td>Não</td></tr><tr><td>rede_2.mat</td><td>Triangle</td><td>Não</td></tr><tr><td>rede_3.mat</td><td>Circle</td><td>Sim</td></tr></table>	rede_1.mat	Triangle	Não	rede_2.mat	Triangle	Não	rede_3.mat	Circle	Sim
rede_1.mat	Triangle	Não									
rede_2.mat	Triangle	Não									
rede_3.mat	Circle	Sim									
	Circle2	<table border="1"><tr><td>rede_1.mat</td><td>Triangle</td><td>Não</td></tr><tr><td>rede_2.mat</td><td>Triangle</td><td>Não</td></tr><tr><td>rede_3.mat</td><td>Triangle</td><td>Não</td></tr></table>	rede_1.mat	Triangle	Não	rede_2.mat	Triangle	Não	rede_3.mat	Triangle	Não
rede_1.mat	Triangle	Não									
rede_2.mat	Triangle	Não									
rede_3.mat	Triangle	Não									
	Circle3	<table border="1"><tr><td>rede_1.mat</td><td>Circle</td><td>Sim</td></tr><tr><td>rede_2.mat</td><td>Circle</td><td>Sim</td></tr><tr><td>rede_3.mat</td><td>Circle</td><td>Sim</td></tr></table>	rede_1.mat	Circle	Sim	rede_2.mat	Circle	Sim	rede_3.mat	Circle	Sim
rede_1.mat	Circle	Sim									
rede_2.mat	Circle	Sim									
rede_3.mat	Circle	Sim									
	Circle4	<table border="1"><tr><td>rede_1.mat</td><td>Circle</td><td>Sim</td></tr><tr><td>rede_2.mat</td><td>Circle</td><td>Sim</td></tr><tr><td>rede_3.mat</td><td>Circle</td><td>Sim</td></tr></table>	rede_1.mat	Circle	Sim	rede_2.mat	Circle	Sim	rede_3.mat	Circle	Sim
rede_1.mat	Circle	Sim									
rede_2.mat	Circle	Sim									
rede_3.mat	Circle	Sim									

	Circle5 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">rede_1.mat</td><td style="padding: 2px;">Triangle</td><td style="padding: 2px;">Não</td></tr> <tr> <td style="padding: 2px;">rede_2.mat</td><td style="padding: 2px;">Triangle</td><td style="padding: 2px;">Não</td></tr> <tr> <td style="padding: 2px;">rede_3.mat</td><td style="padding: 2px;">Triangle</td><td style="padding: 2px;">Não</td></tr> </table>	rede_1.mat	Triangle	Não	rede_2.mat	Triangle	Não	rede_3.mat	Triangle	Não
rede_1.mat	Triangle	Não								
rede_2.mat	Triangle	Não								
rede_3.mat	Triangle	Não								

Desempenho das Redes:

- **Rede 1:** 2/5 acertos (40%)
- **Rede 2:** 2/5 acertos (40%)
- **Rede 3:** 3/5 acertos (60%)

Padrão de Erros:

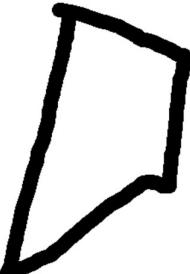
- 60% dos erros classificam círculos como **triângulos**
- **Rede 3** é a menos pior, mas ainda insatisfatória

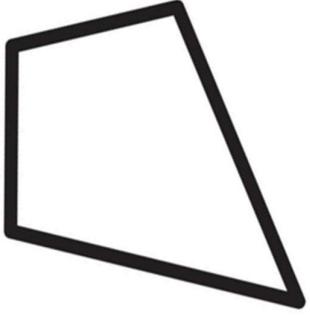
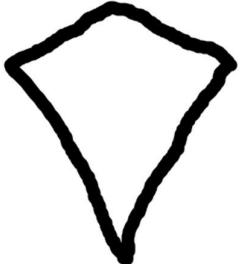
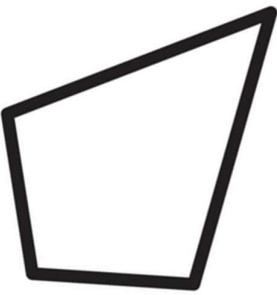
Conclusão:

As redes têm **dificuldade extrema em distinguir círculos de triângulos**, sugerindo:

- Falta de características robustas para formas curvilíneas
- Necessidade de aumentar exemplos de treino com variações de escala/rotação

Kite

	Kite1 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">rede_1.mat</td><td style="padding: 2px;">Square</td><td style="padding: 2px;">Não</td></tr> <tr> <td style="padding: 2px;">rede_2.mat</td><td style="padding: 2px;">Kite</td><td style="padding: 2px;">Sim</td></tr> <tr> <td style="padding: 2px;">rede_3.mat</td><td style="padding: 2px;">Kite</td><td style="padding: 2px;">Sim</td></tr> </table>	rede_1.mat	Square	Não	rede_2.mat	Kite	Sim	rede_3.mat	Kite	Sim
rede_1.mat	Square	Não								
rede_2.mat	Kite	Sim								
rede_3.mat	Kite	Sim								

	<table border="1" data-bbox="714 540 1333 614"> <thead> <tr> <th data-bbox="747 561 817 593">Kite3</th> <th data-bbox="894 551 992 582">rede_1.mat</th> <th data-bbox="1090 551 1144 582">Triangle</th> <th data-bbox="1241 551 1279 582">Não</th> </tr> </thead> <tbody> <tr> <td></td> <td data-bbox="894 582 992 614">rede_2.mat</td> <td data-bbox="1090 582 1144 614">Triangle</td> <td data-bbox="1241 582 1279 614">Não</td> </tr> <tr> <td></td> <td data-bbox="894 614 992 646">rede_3.mat</td> <td data-bbox="1090 614 1144 646">Circle</td> <td data-bbox="1241 614 1279 646">Não</td> </tr> </tbody> </table>	Kite3	rede_1.mat	Triangle	Não		rede_2.mat	Triangle	Não		rede_3.mat	Circle	Não
Kite3	rede_1.mat	Triangle	Não										
	rede_2.mat	Triangle	Não										
	rede_3.mat	Circle	Não										
	<table border="1" data-bbox="747 931 1333 1005"> <thead> <tr> <th data-bbox="747 952 801 984">Kite4</th> <th data-bbox="894 963 992 994">rede_1.mat</th> <th data-bbox="1090 963 1144 994">Kite</th> <th data-bbox="1241 963 1279 994">Sim</th> </tr> </thead> <tbody> <tr> <td></td> <td data-bbox="894 994 992 1026">rede_2.mat</td> <td data-bbox="1090 994 1144 1026">Kite</td> <td data-bbox="1241 994 1279 1026">Sim</td> </tr> <tr> <td></td> <td data-bbox="894 1026 992 1058">rede_3.mat</td> <td data-bbox="1090 1026 1144 1058">Kite</td> <td data-bbox="1241 1026 1279 1058">Sim</td> </tr> </tbody> </table>	Kite4	rede_1.mat	Kite	Sim		rede_2.mat	Kite	Sim		rede_3.mat	Kite	Sim
Kite4	rede_1.mat	Kite	Sim										
	rede_2.mat	Kite	Sim										
	rede_3.mat	Kite	Sim										
	<table border="1" data-bbox="747 1322 1333 1396"> <thead> <tr> <th data-bbox="747 1343 801 1374">Kite5</th> <th data-bbox="894 1353 992 1385">rede_1.mat</th> <th data-bbox="1090 1353 1144 1385">Triangle</th> <th data-bbox="1241 1353 1279 1385">Não</th> </tr> </thead> <tbody> <tr> <td></td> <td data-bbox="894 1385 992 1417">rede_2.mat</td> <td data-bbox="1090 1385 1144 1417">Trapezoid</td> <td data-bbox="1241 1385 1279 1417">Não</td> </tr> <tr> <td></td> <td data-bbox="894 1417 992 1448">rede_3.mat</td> <td data-bbox="1090 1417 1144 1448">Circle</td> <td data-bbox="1241 1417 1279 1448">Não</td> </tr> </tbody> </table>	Kite5	rede_1.mat	Triangle	Não		rede_2.mat	Trapezoid	Não		rede_3.mat	Circle	Não
Kite5	rede_1.mat	Triangle	Não										
	rede_2.mat	Trapezoid	Não										
	rede_3.mat	Circle	Não										

Desempenho das Redes:

- **Rede 1:** 3/5 acertos (60%)
 - **Rede 2:** 4/5 acertos (80%)
 - **Rede 3:** 3/5 acertos (60%)

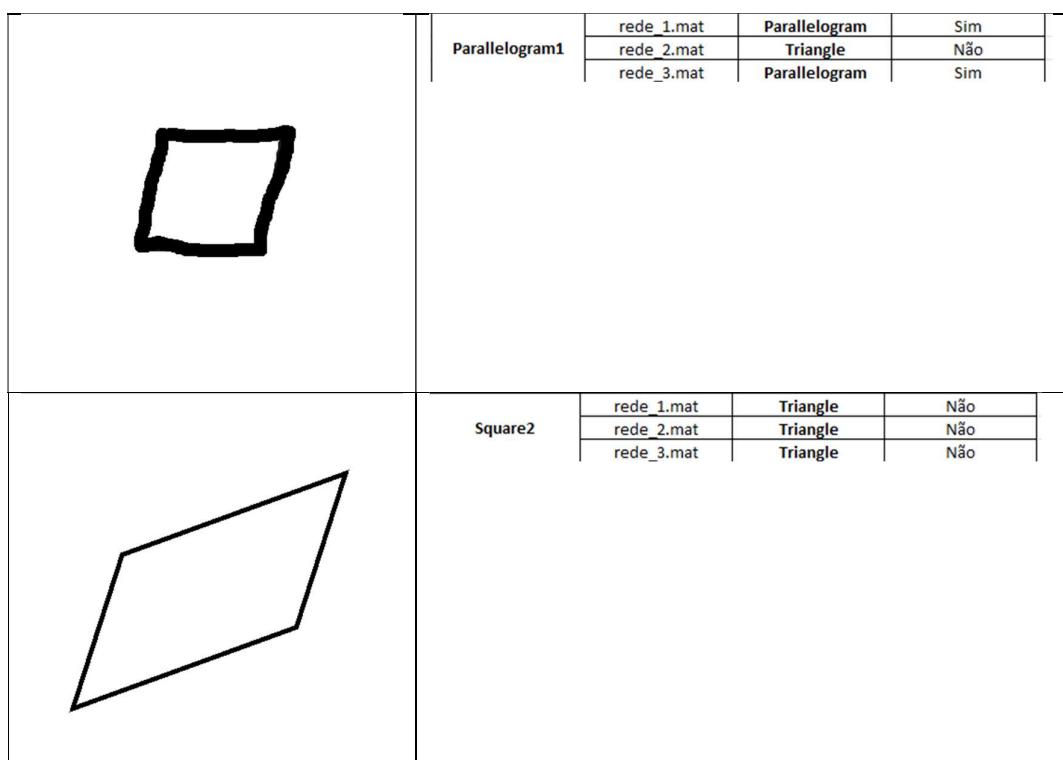
Padrão de Erros:

- Quando falham, classificam como triângulo (60% dos erros)
 - Rede 2 é a mais precisa para esta classe

Conclusão:

- Melhor desempenho geral, mas ainda problemático
 - Erros revelam similaridade percebida com triângulos
 - Priorizar Rede 2 para aplicações com kites

Paralelogramma



Hand-drawn sketches of irregular shapes:

- Top row: A trapezoid-like shape.
- Middle row: A parallelogram-like shape.
- Bottom row: An irregular quadrilateral shape.

Classification results for Square3:

	rede_1.mat	Square	Sim
rede_2.mat	Square	Sim	
rede_3.mat	Square	Sim	

Classification results for Square4:

	rede_1.mat	Triangle	Não
rede_2.mat	Triangle	Não	
rede_3.mat	Triangle	Não	

Classification results for Square5:

	rede_1.mat	Triangle	Não
rede_2.mat	Triangle	Não	
rede_3.mat	Triangle	Não	

Desempenho das Redes:

- **Rede 1:** 2/5 acertos (40%)
- **Rede 2:** 1/5 acertos (20%)
- **Rede 3:** 3/5 acertos (60%)

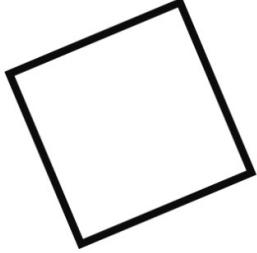
Padrão de Erros:

- 73% dos erros são classificações como **triângulo** ou **círculo**
- **Rede 3** mostra relativa superioridade

Conclusão:

- **Confusão sistemática com formas mais simples** (triângulos)
- Indica que as redes **não aprendem ângulos oblíquos** adequadamente
- Sugestão: Incluir paralelogramos rotacionados no treino

Quadrado

	<table border="1"><thead><tr><th>Square1</th><th>rede_1.mat</th><th>Triangle</th><th>Não</th></tr></thead><tbody><tr><td>rede_1.mat</td><td>Triangle</td><td>Não</td></tr><tr><td>rede_2.mat</td><td>Triangle</td><td>Não</td></tr></tbody></table>	Square1	rede_1.mat	Triangle	Não	rede_1.mat	Triangle	Não	rede_2.mat	Triangle	Não
Square1	rede_1.mat	Triangle	Não								
rede_1.mat	Triangle	Não									
rede_2.mat	Triangle	Não									
											
											
	<table border="1"><thead><tr><th>Square2</th><th>rede_1.mat</th><th>Triangle</th><th>Não</th></tr></thead><tbody><tr><td>rede_1.mat</td><td>Triangle</td><td>Não</td></tr><tr><td>rede_2.mat</td><td>Triangle</td><td>Não</td></tr></tbody></table>	Square2	rede_1.mat	Triangle	Não	rede_1.mat	Triangle	Não	rede_2.mat	Triangle	Não
Square2	rede_1.mat	Triangle	Não								
rede_1.mat	Triangle	Não									
rede_2.mat	Triangle	Não									
	<table border="1"><thead><tr><th>Square3</th><th>rede_1.mat</th><th>Square</th><th>Sim</th></tr></thead><tbody><tr><td>rede_1.mat</td><td>Square</td><td>Sim</td></tr><tr><td>rede_2.mat</td><td>Square</td><td>Sim</td></tr></tbody></table>	Square3	rede_1.mat	Square	Sim	rede_1.mat	Square	Sim	rede_2.mat	Square	Sim
Square3	rede_1.mat	Square	Sim								
rede_1.mat	Square	Sim									
rede_2.mat	Square	Sim									
	<table border="1"><thead><tr><th>Square4</th><th>rede_1.mat</th><th>Triangle</th><th>Não</th></tr></thead><tbody><tr><td>rede_1.mat</td><td>Triangle</td><td>Não</td></tr><tr><td>rede_2.mat</td><td>Triangle</td><td>Não</td></tr></tbody></table>	Square4	rede_1.mat	Triangle	Não	rede_1.mat	Triangle	Não	rede_2.mat	Triangle	Não
Square4	rede_1.mat	Triangle	Não								
rede_1.mat	Triangle	Não									
rede_2.mat	Triangle	Não									

	Square5	rede_1.mat	Triangle	Não
		rede_2.mat	Triangle	Não
		rede_3.mat	Triangle	Não

Desempenho das Redes:

- **Rede 1:** 1/5 acertos (20%)
- **Rede 2:** 1/5 acertos (20%)
- **Rede 3:** 1/5 acertos (20%)

Padrão de Erros:

- **80% dos erros** são classificações como **triângulo**
- Falha generalizada em todas as redes

Conclusão:

- **Problema crítico e consistente**
- As redes **não reconhecem simetria quadrangular**
- Ação urgente: Revisão do pré-processamento ou inclusão de quadrados com ruído/no treino

Trapézio

	Trapezoid1	rede_1.mat	Trapezoid	Sim
		rede_2.mat	Trapezoid	Sim
		rede_3.mat	Parallelogram	Não

	Trapezoid2	rede_1.mat	Triangle	Não
		rede_2.mat	Triangle	Não
		rede_3.mat	Triangle	Não
	Trapezoid3	rede_1.mat	Triangle	Não
		rede_2.mat	Triangle	Não
		rede_3.mat	Triangle	Não
	Trapezoid4	rede_1.mat	Trapezoid	Sim
		rede_2.mat	Kite	Não
		rede_3.mat	Trapezoid	Sim
	Trapezoids5	rede_1.mat	Triangle	Não
		rede_2.mat	Triangle	Não
		rede_3.mat	Triangle	Não

Desempenho das Redes:

- **Rede 1:** 2/5 acertos (100%)
 - **Rede 2:** 1/5 acertos (100%)
 - **Rede 3:** 1/5 acertos (100%)

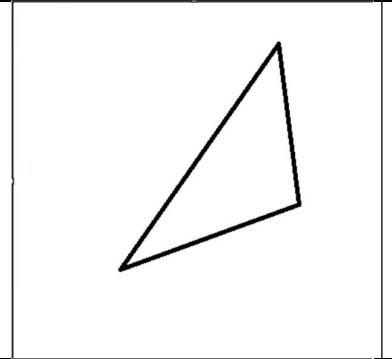
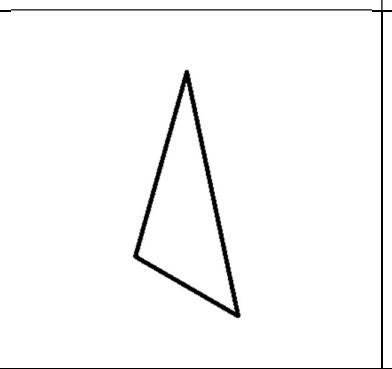
Padrão de Erros:

- **Maior parte dos erros** são classificações como **triângulo**
- Imagem que dependendo da maneira de como é feita pode ser similar a um paralelogramo ou kite

Conclusão

- Sugestão: Incluir mais Trapézios rotacionados e em outras posições no treino

Triangulo

	Triangle1	<table border="1"><tr><td>rede_1.mat</td><td>Triangle</td><td>Sim</td></tr><tr><td>rede_2.mat</td><td>Triangle</td><td>Sim</td></tr><tr><td>rede_3.mat</td><td>Triangle</td><td>Sim</td></tr></table>	rede_1.mat	Triangle	Sim	rede_2.mat	Triangle	Sim	rede_3.mat	Triangle	Sim
rede_1.mat	Triangle	Sim									
rede_2.mat	Triangle	Sim									
rede_3.mat	Triangle	Sim									
	Triangle2	<table border="1"><tr><td>rede_1.mat</td><td>Triangle</td><td>Sim</td></tr><tr><td>rede_2.mat</td><td>Triangle</td><td>Sim</td></tr><tr><td>rede_3.mat</td><td>Triangle</td><td>Sim</td></tr></table>	rede_1.mat	Triangle	Sim	rede_2.mat	Triangle	Sim	rede_3.mat	Triangle	Sim
rede_1.mat	Triangle	Sim									
rede_2.mat	Triangle	Sim									
rede_3.mat	Triangle	Sim									
	Triangle3	<table border="1"><tr><td>rede_1.mat</td><td>Triangle</td><td>Sim</td></tr><tr><td>rede_2.mat</td><td>Triangle</td><td>Sim</td></tr><tr><td>rede_3.mat</td><td>Triangle</td><td>Sim</td></tr></table>	rede_1.mat	Triangle	Sim	rede_2.mat	Triangle	Sim	rede_3.mat	Triangle	Sim
rede_1.mat	Triangle	Sim									
rede_2.mat	Triangle	Sim									
rede_3.mat	Triangle	Sim									

	Triangle4	rede_1.mat	Triangle	Sim
		rede_2.mat	Triangle	Sim
		rede_3.mat	Triangle	Sim
	Triangle5	rede_1.mat	Triangle	Sim
		rede_2.mat	Triangle	Sim
		rede_3.mat	Triangle	Sim

Desempenho das Redes:

- **Rede 1:** 5/5 acertos (100%)
- **Rede 2:** 5/5 acertos (100%)
- **Rede 3:** 5/5 acertos (100%)

Conclusão:

- **As redes conseguem classificar bem imagens triangulares**

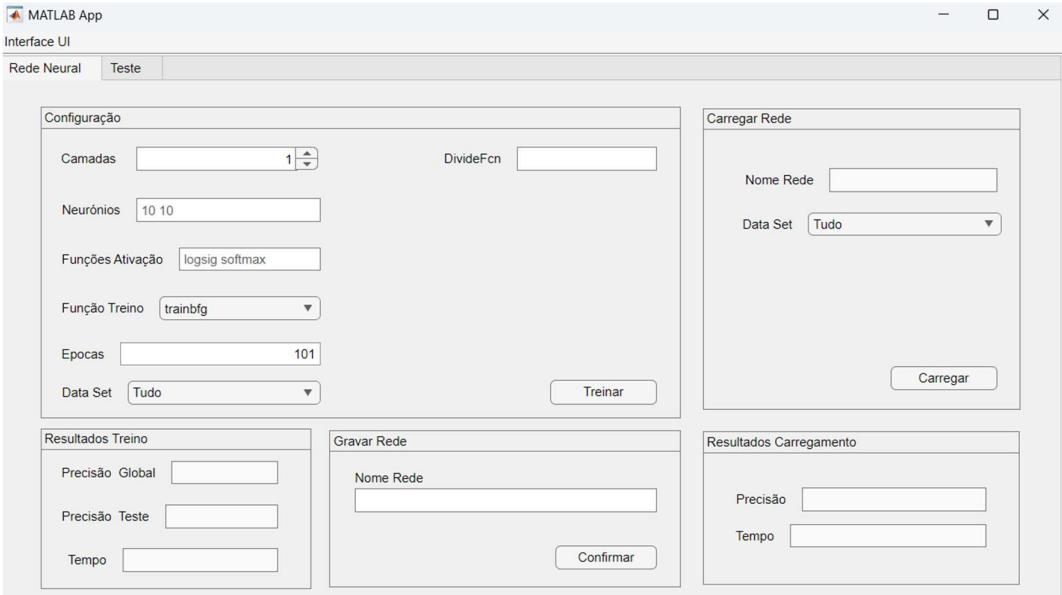
Conclusão Geral

Analizando estas classificações podemos concluir que tanto a grossura da linha, tanto a posição e escala influencia muito na hora da classificação. Com a rede classifica a partir de matrizes binárias estes já referidos influencia a matriz logo como podemos ver nas classificações mesmo a rede tendo boas precisões globais não foi capaz de classificar todas corretamente.

Para resolução deste problema era aumentar o número de imagens no treino das redes com diferentes posições, rotações e grossuras.

4. Aplicação

Rede Neural



No painel “Configuração”, é possível criação da mesma através da personalização dos campos:

- Camadas
- Neurónios
- Funções de Ativação
- Função de Treino
- Épocas
- Data Set
- DivideFcn

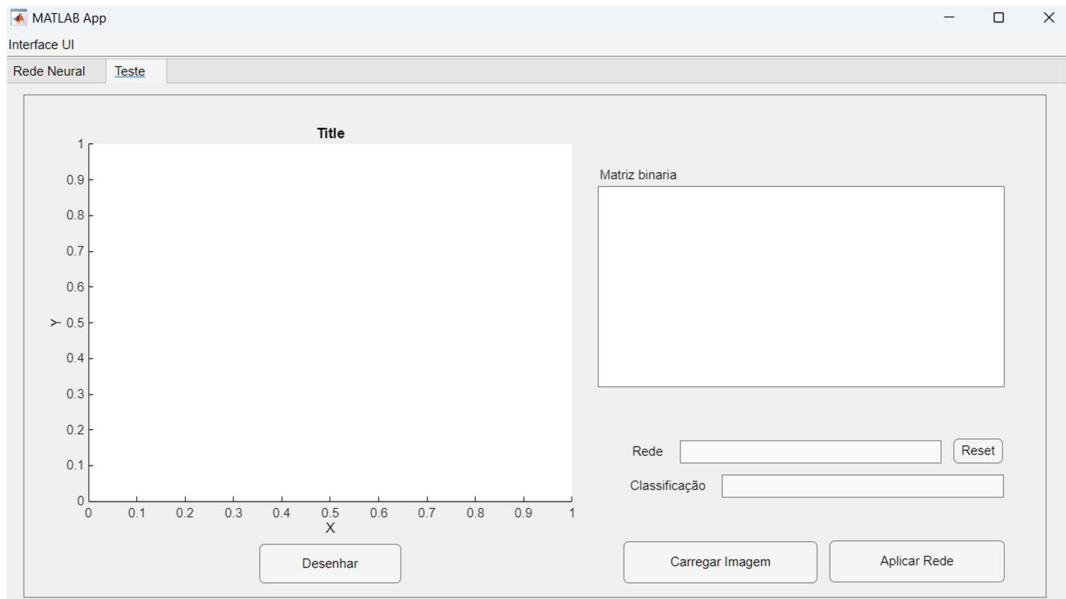
Após carregar no botão “Treinar”, o treino será iniciado e quanto estiver concluído será exibido os resultados no painel “Resultados Treino”, onde exibirá dados sobre a precisão global, precisão teste (se na editField do DivideFcn for preenchida com “dividerand”) e tempo de execução.

No painel “Gravar Rede” é possível salvar a rede treinada, atribuindo um nome, e clicando no botão confirmar a rede será guardada na pasta “Saves”.

No painel “Carregar Rede” é possível carregar a rede guardada anteriormente, onde o nome da rede será atribuído automaticamente com o caminho onde o ficheiro se encontra, e é possível escolher se quer testar nas 3 pastas existentes (Start, Train, Test) em simultâneo (DropDown Tudo) ou apenas em uma(DropDown Pasta).

No painel “Resultados Carregamento” exibe os resultados (precisão e tempo) da rede aplicada ao dataset definido anteriormente.

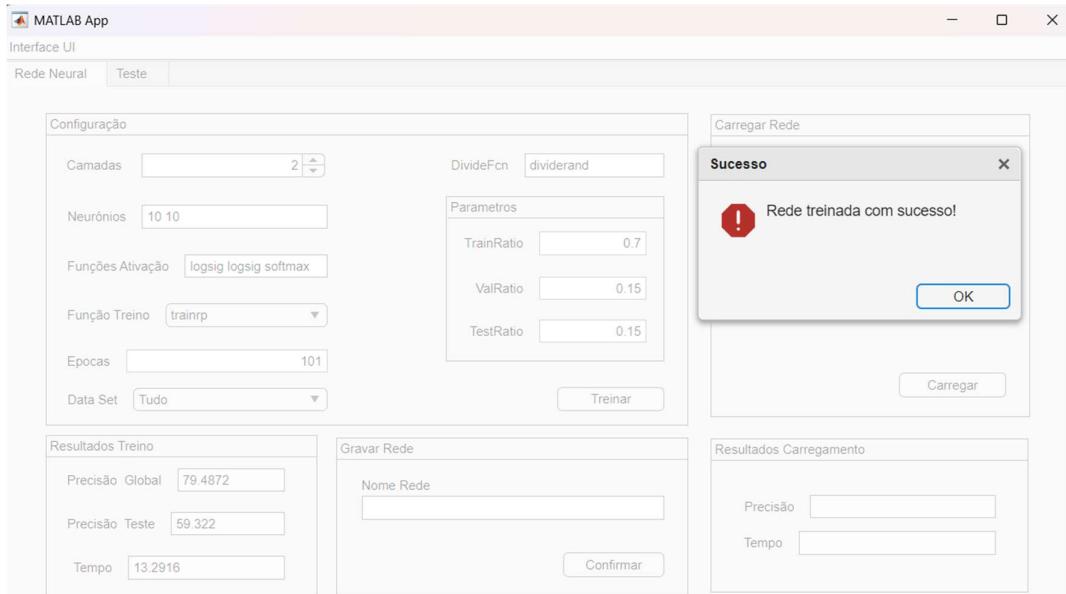
Teste

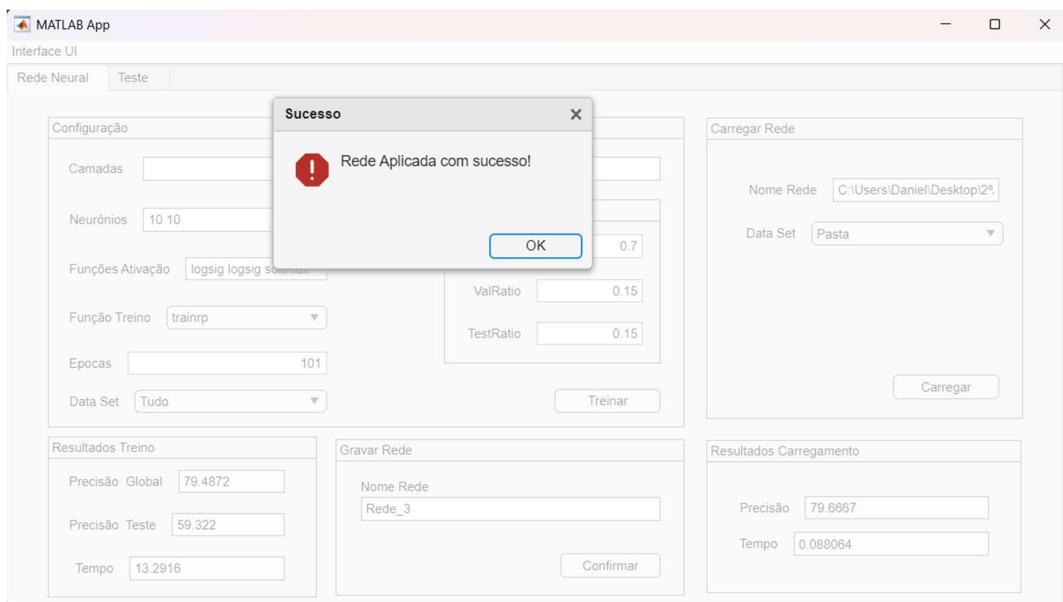
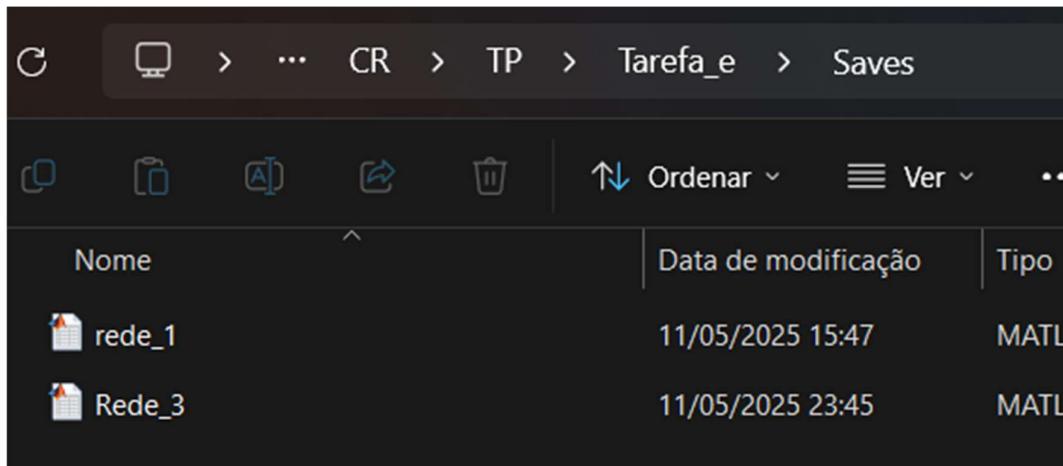
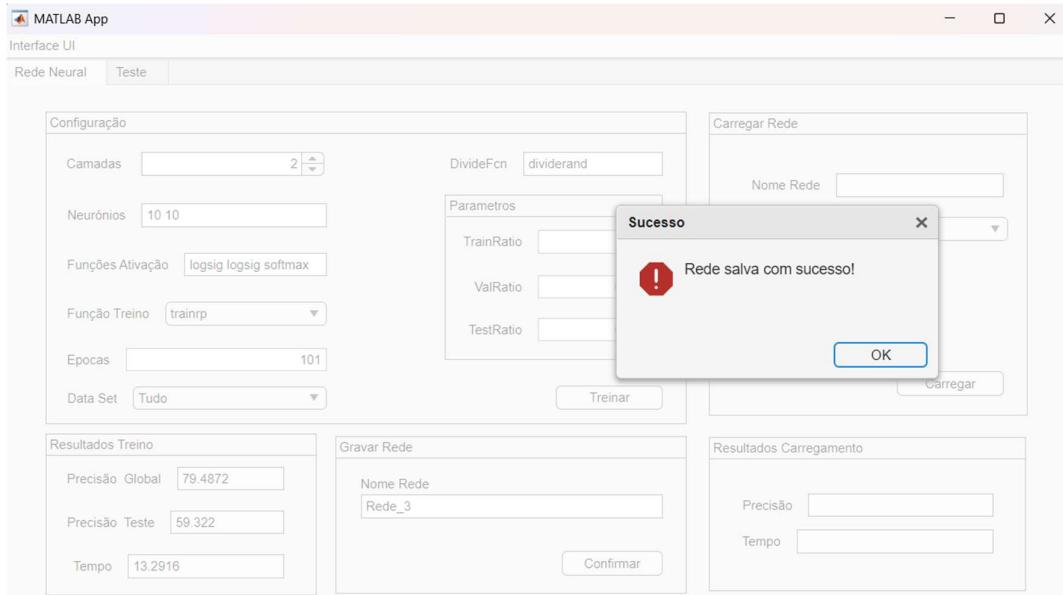


Nesta aba é possível testar uma rede, de forma que acerte a figura desenhada ou uma imagem carregada.

Ao clicar no botão “Desenhar” aparecerá um espaço para desenhar a figura geométrica pretendida, o programa converte o desenho numa matriz binária em seguida clicando no botão “Aplicar Rede” abrirá o explorador de ficheiros onde terá de selecionar a rede pretendida, após a seleção, o campo “Rede” será preenchido automaticamente com o caminho da mesma e exhibirá o resultado da rede no espaço “Classificação”. Assim também acontece quando o utilizador carrega a imagem.

Galeria de Funcionamento





The figure shows the MATLAB App interface for a Neural Network. It includes tabs for 'Rede Neural' and 'Teste'. The main window contains several panels:

- Configuração:** Includes fields for 'Camadas' (2), 'Neurónios' (10 10), 'Funções Ativação' (logsig logsig softmax), 'Função Treino' (trainrp), 'Epochas' (101), and 'Data Set' (Tudo). A 'Treinar' button is located below these.
- Parametros:** Shows 'TrainRatio' (0.7), 'ValRatio' (0.15), and 'TestRatio' (0.15).
- Carregar Rede:** Includes fields for 'Nome Rede' (C:\Users\Daniel\Desktop\20) and 'Data Set' (Pasta), with a 'Carregar' button.
- Resultados Treino:** Displays 'Precisão Global' (79.4872), 'Precisão Teste' (59.322), and 'Tempo' (13.2916).
- Gravar Rede:** Includes a 'Nome Rede' field (Rede_3) and a 'Confirmar' button.
- Resultados Carregamento:** Displays 'Precisão' (79.6667) and 'Tempo' (0.088064).