

Reconhecimento de Faces: um Estudo Integrado de Classificadores, Pré-processamento e Controle de Acesso

Lucas José Lemos Braz

Agosto, 2025

1 Introdução

O texto está organizado da seguinte forma: Seção 2 apresenta o conjunto de dados, as técnicas de pré-processamento e os classificadores avaliados, descrevendo o protocolo experimental que sustenta a análise. A Seção 3 discute os resultados obtidos em cada atividade previstas nas atividades solicitadas: classificação no espaço de pixels (Atividades 1–2), decorrelação com PCA (Atividades 3–4), redução de dimensionalidade (Atividades 5–6), aplicação da transformação Box–Cox (Atividade 7) e controle de acesso (Atividade 8). Finalmente, a Seção 4 sintetiza as conclusões principais, identificando as decisões metodológicas que se mostram importantes para equilibrar desempenho, generalização e eficiência.

2 Metodologia

2.1 Conjunto de Dados e Pré-processamento Inicial

O estudo utiliza o conjunto Yale A, composto por quinze indivíduos fotografados em onze condições de iluminação e pose, totalizando 165 imagens. Para as primeiras etapas, as imagens foram redimensionadas para 30×30 pixels, produzindo vetores de $d = 900$ atributos após a *flatten*. Esse tamanho intermediário foi escolhido após um estudo de escalas (20×20 , 30×30 e 40×40) que mensurou o tempo de treinamento e a acurácia dos classificadores. A figura 1 demonstra que a escala de 30×30 equilibra a preservação de detalhes faciais e o custo computacional ao evitar a perda de textura de resoluções menores e o crescimento de tempo de redes profundas em resoluções maiores.

Para evitar vazamento de dados, todas as normalizações e transformações foram ajustadas exclusivamente sobre o conjunto de treino de cada repetição. Foram testadas três normalizações: `minmax` (escala $[0, 1]$), `minmax_pm1` ($[-1, 1]$) e `zscore` (média zero e desvio padrão unitário), aplicadas após a divisão treino/teste. A PCA foi implementada via decomposição SVD em sua forma compacta: primeiro os dados foram centrados; depois, quando utilizada como rotação, todos os d componentes principais foram mantidos ($q = d$); quando utilizada para redução, apenas os q maiores autovalores foram retidos. A transformação Box–Cox, investigada na Atividade 7, foi aplicada componente a componente apenas após a redução de dimensionalidade; por exigir entradas positivas, um

deslocamento baseado no valor mínimo de cada componente no treino foi somado antes de estimar o parâmetro λ que maximiza a verossimilhança.

2.2 Classificadores Avaliados

Quatro classificadores básicos foram avaliados ao longo das Atividades 1–7: *Mínimos Quadrados* (MQ), um classificador linear de solução analítica; *Perceptron Logístico* (PL), ou regressão *softmax*, treinado com gradiente descendente; e redes neurais *multilayer perceptron* de uma e duas camadas ocultas (MLP-1H e MLP-2H). Para cada modelo (exceto o MQ) foi realizada uma busca aleatória de 200 configurações de hiperparâmetros, variando a normalização, a taxa de aprendizado, o otimizador (SGD, Momentum, Nesterov, RMSProp, Adam), as funções de ativação (*tanh*, sigmoide, ReLU, Leaky ReLU, ReLU6 e Swish), o número de neurônios por camada e o clipping de gradiente. O MQ possui um único hiperparâmetro de regularização L_2 . Cada candidata foi avaliada por dez repetições de validação interna para reduzir a variância na seleção. Uma vez escolhida a configuração ótima, o modelo foi re-ajustado em treino+validação e avaliado em 50 repetições independentes com divisão estratificada (80% treino, 20% teste). Essa estratificação garante que a proporção de imagens por sujeito seja preservada, diminuindo o risco de sobreajuste à distribuição de um indivíduo específico.

Na Atividade 8 foram adicionados classificadores destinados à detecção de anomalias: autoencoders com uma e duas camadas ocultas (AE-1H e AE-2H), One-Class SVM e Isolation Forest. Esses modelos foram configurados para aprender a distribuição das faces autorizadas e identificar como intrusos as amostras que não se ajustarem ao padrão de normalidade.

Para as atividades que envolveram retreinamento dos modelos (Atividades 1–7), definiu-se um espaço de busca de hiperparâmetros abrangente. A Tabela 1 sintetiza os valores considerados para cada classificador: incluem-se diferentes níveis de regularização L_2 , taxas de aprendizado, números de épocas, otimizadores, funções de ativação, tamanhos de camadas ocultas e limites de clipping de gradiente. Para o problema de controle de acesso (Atividade 8) adotou-se um espaço semelhante, porém o conjunto de combinações possíveis foi restringido a fim de manter o foco na comparação conceitual entre as duas abordagens.

Tabela 1: Espaço de busca dos hiperparâmetros utilizado nas Atividades 1–7.

Classificador	Hiperparâmetros e valores considerados
MQ	$l_2 \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$
PL	$lr \in \{5 \times 10^{-3}, 10^{-2}, 2 \times 10^{-2}\}$; $epochs \in \{100, 200, 300\}$; $l_2 \in \{0, 10^{-4}, 10^{-3}\}$; $opt \in \{sgd, momentum, nesterov, rmsprop, adam\}$
MLP-1H	$hidden \in \{4, 8, 16, 32, 64, 128, 256, 512\}$; $activation \in \{tanh, sigmoid, relu, leaky_relu, relu6, swish\}$; $lr \in \{5 \times 10^{-3}, 10^{-2}, 2 \times 10^{-2}\}$; $epochs \in \{150, 200, 300\}$; $l_2 \in \{0, 10^{-4}, 10^{-3}\}$; $opt \in \{sgd, momentum, nesterov, rmsprop, adam\}$; $clip_grad \in \{0, 2, 5, 10\}$
MLP-2H	$hidden \in \{(h_1, h_2) \mid h_1, h_2 \in \{4, 8, 16, 32, 64, 128, 256, 512\}\}$; mesmo espaço de $activation, lr, epochs, l_2, opt$ e $clip_grad$ do MLP-1H

Um detalhe de implementação importante para o MQ é a forma como se resolve o problema linear. Na ausência de regularização, a solução analítica recorre à pseudo-inversa de Moore–Penrose para obter a matriz de pesos. Com regularização L_2 , a solução é obtida resolvendo o sistema linear $(X^T X + \lambda I)W = X^T Y$, o que melhora o condicionamento numérico e evita problemas com matrizes singulares.

Para além das funções usuais de ativação sigmoidal e tangente hiperbólica, foram testadas ativações modernas. A *Leaky ReLU* mantém um gradiente não nulo para entradas negativas, reduzindo o problema de neurônios inativos. A *ReLU6* é uma ReLU limitada ao intervalo $[0, 6]$, evitando que a saída cresça indefinidamente e sendo útil em modelos compactos. A função *Swish* é definida por $\text{swish}(x) = x \sigma(x) = x/(1 + e^{-x})$; essa ativação suave permite que valores negativos moderados contribuam para a saída, facilitando a transição em torno de zero. A escolha da ativação interage com a normalização e o otimizador e foi tratada como parte do espaço de busca.

2.3 Pipeline Analítico

As atividades solicitadas foram executadas de forma encadeada, com cada fase alimentando a seguinte. Nas duas primeiras atividades trabalhou-se diretamente no espaço de pixels: as imagens vetorizadas foram normalizadas conforme a estratégia selecionada na busca de hiperparâmetros, e os classificadores foram treinados a partir dos 900 atributos originais. Esse ciclo estabeleceu uma linha de base de desempenho sem pré-processamento linear, permitindo avaliar o impacto de outras transformações.

Nas atividades 3–4 introduziu-se a rotação da base por PCA sem redução dimensional ($q = d$). Essa transformação descorrela os atributos e melhora o condicionamento numérico para otimização. Manter todos os componentes preserva a informação e, a priori, esperava-se apenas uma aceleração dos treinos. O experimento revelou até que ponto a rotação altera o comportamento dos modelos.

A quinta e a sexta atividades focaram na redução de dimensionalidade. Para determinar a dimensão intrínseca do conjunto, estimou-se a variância explicada cumulativa do PCA aplicado ao treino completo e observou-se que 98% da variância está concentrada nos 79 primeiros componentes. A projeção para esse subespaço de 79 dimensões, seguida da normalização escolhida, foi então utilizada em cada repetição; o PCA foi sempre ajustado apenas nos dados de treino para evitar vazamento. Os *eigenfaces* resultantes foram usados para projetar treino e teste, permitindo medir o ganho de desempenho com a compressão.

A sétima atividade acrescentou uma transformação não linear: após a projeção para $q = 79$, aplicou-se a Box–Cox a cada componente principal, estimando o parâmetro λ no treino e garantindo a positividade. Em seguida, uma normalização Z-Score preparou os dados para o classificador. Esta etapa investigou se a aproximação a uma distribuição Gaussiana poderia beneficiar classificadores lineares ou redes rasas.

Finalmente, compararam-se duas formulações para o problema de controle de acesso diante da presença de um intruso. Na abordagem supervisionada, o intruso é tratado como uma 16^a classe, e os classificadores são treinados no pipeline PCA ($q = 79$) \rightarrow Box–Cox \rightarrow Z–Score \rightarrow classificador.

Já na abordagem não supervisionada, modela-se apenas a distribuição das faces autorizadas, empregando-se os percentis dos escores de anomalia para definir limiares de rejeição. Enquanto a formulação supervisionada busca maximizar a sensibilidade na detecção do intruso previamente visto, a detecção de anomalias prioriza a redução da taxa de falsos negativos (intrusos aceitos) em relação aos falsos positivos (usuários legítimos rejeitados). A inclusão de onze imagens do intruso no conjunto de testes permitiu simular de forma realista a presença dessa classe nas avaliações.

3 Resultados e Discussão

Atividades 1–2: Classificação no Espaço de Pixels

A Tabela 2 apresenta as estatísticas de acurácia e tempo de execução obtidas pelos quatro classificadores quando treinados diretamente nos vetores de 900 pixels (escala 30×30). Todos os valores são médias sobre 50 repetições com desvio padrão. Observa-se que o classificador de Mínimos Quadrados (MQ) apresentou a melhor acurácia média e o menor desvio padrão, contrariando a intuição de que redes neurais profundas sempre superariam modelos lineares. As MLPs, embora capazes de aprender fronteiras não lineares, sofreram maior variabilidade e não ultrapassaram o desempenho lineares, fenômeno compatível com o regime de poucas amostras por classe do conjunto Yale A. Além disso, o tempo de treino cresce drasticamente à medida que se adicionam camadas e neurônios.

Tabela 2: Atividades 1–2 — Desempenho sem PCA ($d = 900$). Os valores são médias de 50 repetições.

Classificador	Média	Min	Max	Med	DP	Tempo Total (ms)
MQ	0.965	0.911	1.000	0.978	0.024	8.516
PL	0.922	0.844	1.000	0.933	0.033	38.442
MLP-1H	0.928	0.844	0.978	0.933	0.039	252.304
MLP-2H	0.930	0.800	1.000	0.933	0.039	942.703

Tabela 3: Hiperparâmetros selecionados nas Atividades 1–2 (sem PCA).

Classificador	Escala	Normalização	Otimizador	Ativação	Hidden	LR	Épocas	L_2	Clip
MQ	30×30	–	–	–	–	–	–	0.0000	–
PL	30×30	minmax	adam	–	–	0.005	200	0.0001	–
MLP-1H	30×30	minmax_pm1	rmsprop	sigmoid	(128,)	0.005	200	0.0000	2.00
MLP-2H	30×30	minmax	adam	tanh	(256,64)	0.005	300	0.0001	5.00

Os valores da Tabela 3 sintetizam os hiperparâmetros resultantes da busca para as Atividades 1–2. Nota-se que o MQ prescinde de normalização e se beneficia de regularização nula, enquanto o PL necessita de normalização minmax e se ajustou com taxa de aprendizado e regularização moderadas. Para as MLPs, a configuração ótima recorreu a ativação sigmoide ou *tanh* e a tamanhos de camadas relativamente grandes (128 e 256 neurônios), além de técnicas como clipping de gradiente para estabilizar o treinamento, refletindo a maior complexidade dessas redes.

A Figura 1 complementa esses resultados, mostrando o crescimento do tempo de processamento em função da resolução. A escala 30×30 foi selecionada por equilibrar fidelidade visual e custo, permitindo que as atividades subsequentes fossem conduzidas com repetição estatística robusta.

Atividades 3–4: PCA como Rotação ($q = d$)

Ao aplicar o PCA com $q = d$, os dados são rotacionados para um sistema de coordenadas ortogonais sem perda de informação. A Tabela 4 mostra que essa decorrelação reduziu drasticamente o tempo de treinamento de todos os classificadores baseados em gradiente

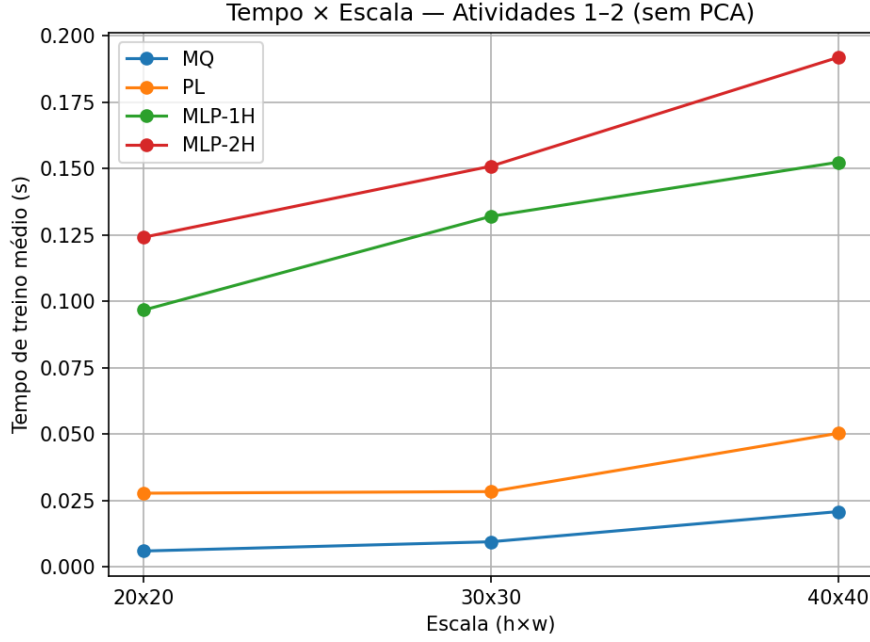


Figura 1: Tempo médio de processamento em função da resolução das imagens nas Atividades 1–2. O custo das MLPs cresce superlinearmente com a dimensionalidade, justificando a escolha da resolução intermediária.

(PL e MLPs). Entretanto, observou-se uma queda de acurácia, especialmente nos modelos não lineares. O MQ, cuja solução é analítica, manteve praticamente o desempenho, sugerindo que a rotação não alterou a separabilidade linear das classes. Para as MLPs, a hipótese de que a decorrelação facilitaria a otimização não se traduziu em ganhos de acurácia; ao contrário, a projeção desalinhou direções naturais de separação, tornando mais difícil a aprendizagem de fronteiras não lineares.

Tabela 4: Atividades 3–4 — Desempenho com PCA como rotação ($q = d = 900$).

Classificador	Média	Min	Max	Med	DP	Tempo Total (ms)
MQ	0.961	0.889	1.000	0.978	0.028	2.258
PL	0.867	0.778	0.933	0.867	0.037	29.576
MLP-1H	0.826	0.644	0.956	0.822	0.062	109.268
MLP-2H	0.840	0.689	0.956	0.822	0.053	299.309

Os valores da Tabela 5 mostram que a busca de hiperparâmetros sob a rotação do PCA optou por normalizações similares às da fase sem PCA, porém reduziu o tamanho das redes neurais: a MLP-1H passou a ter 64 neurônios, e a MLP-2H 128 e 32 neurônios. Além disso, todos os modelos baseados em gradiente utilizaram regularização L_2 muito pequena, indicando que a decorrelação já confere certa robustez ao ajuste e que a simplificação do modelo é suficiente para estabilizar o treino.

Tabela 5: Hiperparâmetros selecionados nas Atividades 3–4 (PCA sem redução).

Classificador	Escala	q	Normalização	Otimizador	Ativação	Hidden	LR	Épocas	L_2	Clip
MQ	30×30	–	none	–	–	–	–	–	0.0000	–
PL	30×30	–	minmax	sgd	–	–	0.0050	200	0.0001	–
MLP-1H	30×30	–	minmax_pm1	rmsprop	sigmoid	(64,)	0.0050	200	0.0001	2.00
MLP-2H	30×30	–	minmax	rmsprop	tanh	(128,32)	0.0050	300	0.0001	5.00

Atividades 5–6: Redução de Dimensionalidade com PCA ($q = 79$)

O estudo da variância explicada cumulativa indicou que cerca de 98% da informação das imagens está contida nos 79 primeiros componentes principais (Figura 2). Esse resultado confirma a forte redundância intrínseca em imagens de faces controladas e motiva a compressão para $q = 79$. Visualmente, as chamadas *eigenfaces* iniciais capturam gradações de iluminação e, somente após as primeiras direções, traços faciais mais finos se tornam visíveis (Figura 3). A Tabela 6 revela que a projeção para 79 dimensões reduziu o custo de treino em ordens de grandeza e, simultaneamente, recuperou ou mesmo melhorou a acurácia dos classificadores em relação ao espaço original. O PL e a MLP-1H igualaram a performance do MQ, sugerindo que os 2% de variância descartados continham principalmente ruído. Embora a MLP-2H continue mais lenta, seus resultados são comparáveis ao MQ, indicando que arquiteturas profundas não são necessárias quando o espaço é adequadamente compactado.

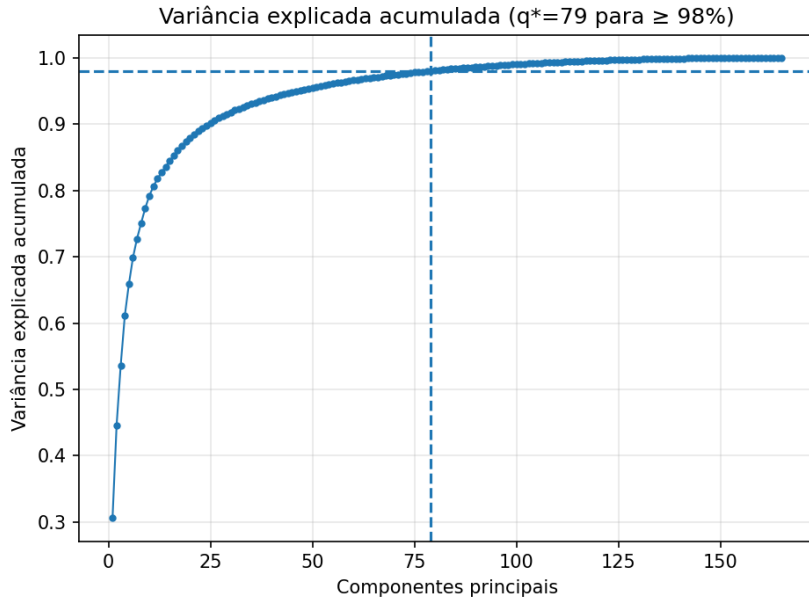


Figura 2: Variância explicada acumulada em função do número de componentes principais. A linha tracejada destaca $q = 79$, ponto a partir do qual mais de 98% da variância total é preservada.

Os hiperparâmetros da Tabela 7 evidenciam que a redução para 79 componentes permitiu um redesenho dos modelos: a normalização passou a ser *z-score* para todas as MLPs, a MLP-1H encolheu para apenas 16 neurônios e adotou a ativação *swish*, enquanto a MLP-2H expandiu a primeira camada para 512 neurônios e utilizou a ativação *leaky_relu*. Esses ajustes mostram que, em um espaço comprimido, vale a pena explo-

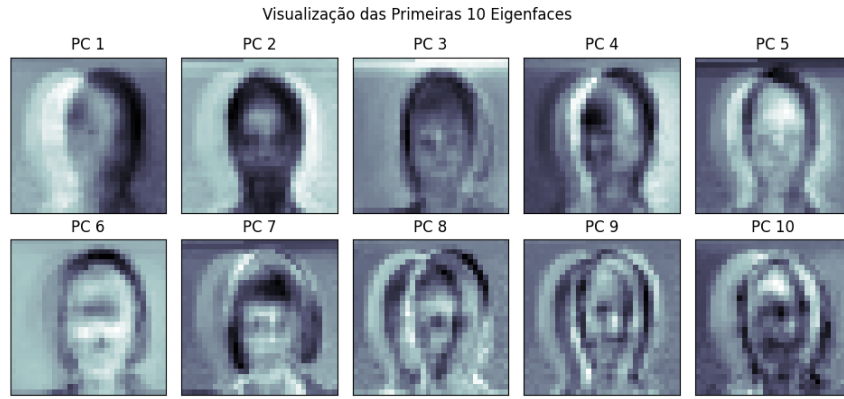


Figura 3: As dez primeiras *eigenfaces* obtidas pelo PCA. As primeiras componentes correspondem a variações globais de iluminação; componentes posteriores capturam contornos faciais comuns.

Tabela 6: Atividades 6 — Desempenho com PCA reduzida ($q = 79$).

Classificador	Média	Min	Max	Med	DP	Tempo Total (ms)
MQ	0.959	0.889	1.000	0.956	0.029	0.260
PL	0.959	0.889	1.000	0.956	0.029	21.692
MLP-1H	0.956	0.889	1.000	0.956	0.027	53.646
MLP-2H	0.948	0.844	1.000	0.956	0.034	442.021

rar funções de ativação suaves e diferentes combinações de regularização para extrair o máximo de discriminabilidade.

Atividade 7: PCA com Box–Cox

A aplicação da transformação Box–Cox após a redução de dimensionalidade pretendia "gaussianizar" os 79 componentes principais, partindo da hipótese de que classificadores lineares beneficiariam-se de entradas mais próximas de uma distribuição normal. Contudo, o Teorema do Limite Central indica que cada componente principal já é uma combinação de centenas de pixels, e portanto tende naturalmente à gaussianidade. A Figura ?? confirma empiricamente que a transformação Box–Cox não apenas não contribuiu para aumentar a acurácia, como produziu uma degradação sistemática em todos os classificadores testados. O custo de ajustar λ para cada componente em cada repetição não se traduziu em melhorias, reforçando que técnicas de pré-processamento devem ser aplicadas com base em análises específicas dos dados e não como receitas universais.

Atividade 8: Controle de Acesso

O desafio final foi construir um mecanismo de controle de acesso capaz de reconhecer usuários autorizados e rejeitar intrusos. Duas filosofias foram comparadas. Na *abordagem supervisionada* (binária), o intruso é considerado uma 16^a classe vista durante o treino; na *abordagem não supervisionada* (unária), o sistema aprende somente a "normalidade" das

Tabela 7: Hiperparâmetros selecionados nas Atividades 5–6 (PCA com redução).

Classificador	Escala	q	Normalização	Otimizador	Ativação	Hidden	LR	Épocas	L_2	Clip
MQ	30×30	79	minmax	–	–	–	–	–	0.0001	–
PL	30×30	79	zscore	sgd	–	–	0.0050	200	0.0001	–
MLP-1H	30×30	79	zscore	rmsprop	swish	(16,)	0.0200	200	0.0001	2.00
MLP-2H	30×30	79	zscore	rmsprop	leaky_relu	(512,64)	0.0050	300	0.0010	0.00

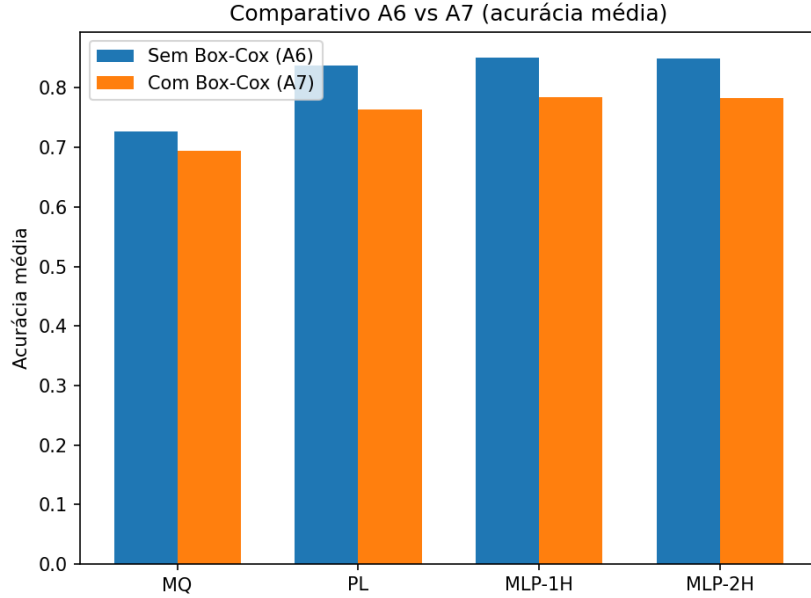


Figura 4: Comparativo entre melhor modelo com e sem Box-Cox

faces autorizadas e rejeita o que se afasta desse padrão. Para ambas, o pipeline de pré-processamento consistiu em PCA com $q = 79$, transformação Box-Cox e normalização Z-Score.

A Tabela 8 resume as médias e desvios padrão de acurácia, taxa de falsos negativos (FNR), taxa de falsos positivos (FPR), sensibilidade (*recall*) e precisão para quatro classificadores. A métrica crítica em segurança é a FNR (probabilidade de aceitar um intruso). Os resultados mostram FNR praticamente nula e sensibilidade unitária para MQ e MLP-1H, dando a falsa impressão de um sistema perfeito. Entretanto, essa performance decorre do fato de o modelo treinar explicitamente com o intruso conhecido: ele apenas memoriza as características daquela pessoa específica. Em ambientes reais, onde intrusos diferentes podem aparecer, esse sistema não generaliza e torna-se inseguro. A precisão modesta (por volta de 0.38) e a FPR acima de 10% indicam que muitos usuários legítimos seriam barrados se o limiar de decisão fosse ajustado para alcançar FNR zero.

A escolha dos hiperparâmetros na Tabela 9 reflete a adaptação dos modelos ao cenário de controle de acesso supervisionado. O MQ foi regulado com regularização $L_2 = 0,001$ para reduzir oscilações, enquanto o PL recorreu ao otimizador RMSProp com taxa de aprendizado relativamente elevada. As MLPs receberam números moderados de neurônios e ativações não lineares distintas, com clipping de gradiente apenas quando necessário, equilibrando capacidade representacional e estabilidade de treino.

Na abordagem unária, somente as imagens dos usuários autorizados foram usadas para treinar um modelo de normalidade. O limiar de decisão foi definido por percentis dos scores de anomalia no conjunto de treino, sem qualquer informação sobre o intruso. A

Tabela 8: Atividade 8 — Métricas de controle de acesso na abordagem binária (supervisionada). Valores médios \pm desvio padrão.

Classificador	Acurácia	FNR	FPR	Sensibilidade	Precisão
MQ	0.886 ± 0.052	0.000 ± 0.000	0.122 ± 0.055	1.000 ± 0.000	0.387 ± 0.119
PL	0.877 ± 0.055	0.107 ± 0.205	0.124 ± 0.056	0.893 ± 0.205	0.355 ± 0.135
MLP-1H	0.886 ± 0.052	0.000 ± 0.000	0.122 ± 0.055	1.000 ± 0.000	0.387 ± 0.119
MLP-2H	0.884 ± 0.054	0.033 ± 0.137	0.122 ± 0.055	0.967 ± 0.137	0.379 ± 0.130

Tabela 9: Hiperparâmetros selecionados na abordagem supervisionada da Atividade 8.

Modelo	Escala	q	Hidden	Ativação	Otimizador	LR	Épocas	L_2	Clip
MQ	30×30	79	—	—	—	—	—	0.001	—
PL	30×30	79	—	—	rmsprop	0.01	200	0	—
MLP-1H	30×30	79	(64,)	leaky_relu	adam	0.02	300	0.001	2
MLP-2H	30×30	79	(512,256)	relu6	rmsprop	0.02	300	0.001	0

Tabela 10 mostra que, como esperado, a FNR é maior do que na abordagem binária, pois o intruso nunca é visto durante o treino. Ainda assim, métodos como o autoencoder de uma camada (AE-1H) e o Isolation Forest alcançaram precisão superior a 0,80 com FPRs inferiores a 6%, demonstrando boa capacidade de rejeitar intrusos não vistos. Os baixos FNRs do One-Class SVM e do AE-2H acontecem às custas de uma FPR excessivamente alta: esses modelos rotularam praticamente todos os usuários como intrusos, ilustrando o perigo do sobreajuste em espaços de alta dimensionalidade mesmo após a redução com PCA.

Tabela 10: Atividade 8 — Métricas de controle de acesso na abordagem unária (detecção de anomalias). Valores médios \pm desvio padrão.

Classificador	Acurácia	FNR	FPR	Sensibilidade	Precisão
PCA_Baseline	0.748 ± 0.044	0.753 ± 0.136	0.031 ± 0.041	0.247 ± 0.136	0.773 ± 0.274
AE_1H	0.799 ± 0.051	0.529 ± 0.143	0.057 ± 0.046	0.471 ± 0.143	0.801 ± 0.152
AE_2H	0.367 ± 0.037	0.000 ± 0.000	0.912 ± 0.053	1.000 ± 0.000	0.326 ± 0.013
OneClassSVM	0.369 ± 0.035	0.000 ± 0.000	0.908 ± 0.050	1.000 ± 0.000	0.327 ± 0.012
IsolationForest	0.780 ± 0.047	0.613 ± 0.130	0.047 ± 0.050	0.387 ± 0.130	0.819 ± 0.171

Os hiperparâmetros na Tabela 11 revelam que os autoencoders necessitam de arquiteturas simétricas e otimizadores com momento para reconstruir faces normais, enquanto os métodos baseados em fronteira (SVM de uma classe) ou em florestas de isolamento dependem de poucos parâmetros, como o número de estimadores ou os valores ν e γ que controlam a tolerância ao erro e a largura do kernel.

Conceitualmente, as diferenças entre as abordagens supervisionada e não supervisionada podem ser sintetizadas pela Tabela 12. A formulação binária visa maximizar o F1-score do intruso visto, mas falha completamente diante de novos intrusos. Já a detecção de anomalias, embora apresente FNR mais elevado, corresponde à prática recomendada em sistemas de segurança: modelar apenas o comportamento normal e usar limiares de rejeição calibrados, eventualmente combinando múltiplos detectores e recorrendo a verificações adicionais (como autenticação de dois fatores) nos casos fronteirços.

Tabela 11: Hiperparâmetros selecionados na abordagem não supervisionada da Atividade 8.

Modelo	Hidden	Ativação	LR	Épocas	L_2	Otimizador	Clip	ν	γ	n_{est}	Seed
PCA_Baseline	–	–	–	–	–	–	–	–	–	–	–
AE_1H	(24,)	tanh	0.005	200	0.0001	nesterov	5.0	–	–	–	–
AE_2H	(123,49,123)	tanh	0.01	200	0.0001	nesterov	5.0	–	–	–	–
OneClassSVM	–	–	–	–	–	–	–	0.05	0.1	–	–
IsolationForest	–	–	–	–	–	–	–	–	–	200	42

Tabela 12: Contraste entre as abordagens supervisionada (binária) e não supervisionada (unária) para controle de acesso.

Característica	Abordagem Binária	Abordagem Unária
Hipótese	O intruso é uma classe conhecida.	O intruso é uma anomalia desconhecida.
Dados de treino	Autorizados + intruso específico	Apenas autorizados
Tarefa	Separar classes (discriminativa)	Modelar a normalidade (generativa/fronteira)
Generalização	Baixa: falha com intrusos não vistos.	Alta: potencial para detectar qualquer intruso.
Validade no mundo real	Fraca: sistema quebradiço e inseguro.	Forte: abordagem padrão em segurança.

Para documentação, o relatório reserva espaço para a inserção manual de nove das onze imagens utilizadas como intruso, distribuídas nas Figuras 5–???. Cada figura deve mostrar uma fotografia distinta do intruso. Mantiveram-se duas imagens fora do relatório para economia de espaço e por serem redundantes.



Figura 5: Imagens do intruso.

4 Conclusão

Ao longo deste estudo foi construída uma narrativa experimental que justificou cada decisão metodológica. Iniciou-se com a análise de escalas e a constatação de que a resolução 30×30 é suficiente para preservar informação facial relevante sem inviabilizar o treinamento repetido de classificadores. O classificador de Mínimos Quadrados estabeleceu um

referencial forte, superando redes neurais em acurácia e estabilidade quando os dados são utilizados diretamente no espaço de pixels. A descorrelação via PCA (Atividades 3–4) acelerou os treinos, mas evidenciou que a rotação não altera o poder discriminativo intrínseco do problema.

O passo decisivo foi a redução de dimensionalidade (Atividades 5–6): comprimir 900 atributos em 79 preservou mais de 98% da variância e recuperou ou melhorou a acurácia, mostrando que a maior parte da estrutura discriminativa das faces reside em um subespaço de baixa dimensão. Tal compressão reduziu o risco de sobreajuste das MLPs e permitiu que modelos lineares e redes rasas tivessem desempenho semelhante. Em contraste, a transformação Box–Cox testada na Atividade 7 ilustrou o risco de aplicar "boas práticas" sem análise prévia: embora concebida para aproximar distribuições de Gauss, ela distorceu um espaço já bem comportado e degradou os resultados.

No cenário aplicado de controle de acesso (Atividade 8), a comparação entre as abordagens supervisionada e unária traz uma lição central. Treinar com um intruso específico leva a taxas de falsos negativos nulas para aquele indivíduo, mas não oferece proteção contra intrusos desconhecidos. A detecção de anomalias, embora apresente FNR maior, proporciona a única estratégia capaz de generalizar para intrusos arbitrários. Em implantação, recomenda-se calibrar limiares com base na tolerância ao risco, adotar verificações complementares (como autenticação de dois fatores) para casos próximos ao limiar e combinar detectores (autoencoder + Isolation Forest) quando necessário.

Em síntese, o relatório demonstra que pré-processamento e escolha metodológica são mais determinantes para o sucesso de um sistema de reconhecimento de faces do que a complexidade do classificador em si. A história construída nestas páginas ilustra como cada decisão, ancorada em evidências quantitativas e justificativas teóricas, contribui para um equilíbrio entre desempenho, robustez e eficiência computacional.