



SCC0276

Aprendizagem de Máquina

**Prof. Dra. Roseli Aparecida Romero
Felipe Padula Sanches**

Alexandre Batistella Bellas, 9763168
Moisés Botarro Ferraz Silva, 8504135

1. Descrição em alto nível da organização do código

A fim de melhor organizar o código, 3 módulos foram criados de acordo com as funcionalidades a serem desenvolvidas:

- ***dataset***

Contém funções relacionadas à leitura de imagens, construção de imagens artificiais utilizando as operações de transformação e armazenamento de imagens no sistema de arquivos.

- ***features***

Apresenta funções para extrair o HOG (Histogram of Oriented Gradients) de uma base de dados.

- ***classification***

Possui funções para determinar os parâmetros que geram os melhores modelos dos classificadores KNN e MLP. Essas funções geram gráficos e tabelas comparando a acurácia para diferentes modelos. O módulo ainda contém uma função para calcular as estatísticas (matriz de confusão, acurácia e precisão) para um modelo passado por parâmetro. Por fim, a função *get_PCA* retorna o vetor de features em uma nova base com menor dimensão, contendo, no mínimo, 50% da variância dos dados originais.

O arquivo *main.py* contém o script principal responsável por chamar as operações corretas para a análise das duas bases de dados consideradas (ORL e Pessoas ICMC) pré-processando-as, treinando os modelos considerados e reportando as acurácias de cada modelo executado.

2. Operações utilizadas para Data Augmentation

Para cada imagem utilizada na base de dados Pessoas ICMC, foram utilizadas as operações que estão presentes no arquivo *dataset.py*, respectivamente:

- Inserção de 10% de ruído com o método *random_noise* no modo *pepper*;
- Inversão de cores (aplicação do efeito negativo);
- Rotação da imagem original num ângulo aleatório entre -40 e 40 graus;
- Ajuste logarítmico;
- Mudança de contraste;
- Correção sigmóide;
- Espelhamento horizontal;
- Rotação da imagem espelhada horizontalmente num ângulo aleatório entre -40 e 40 graus;
- Aplicação de *blur* (desfoco).

Para a base de dados ORL, não foram utilizadas operações de Augmentation.

Nos momentos em que foram aplicadas as rotações, os limites da imagem não preenchem completamente a imagem rotacionada, causando um preenchimento com a cor preta nos cantos da imagem, dando contraste e interferindo no cálculo do HOG, assim como mostra a figura 1.



Figura 1 - Exemplo da falta de preenchimento no momento da rotação da imagem, criando valores inválidos no cálculo do HOG

Assim, para preencher o contorno da imagem, foi utilizado o atributo *mode* da função *rotate* com o valor *edge*, que obtém os pixels dos limites da imagem rotacionada e os replica até o fim da imagem final. Dessa forma, evita-se o surgimento de gradientes que não correspondem a contornos da imagem real durante o cálculo do HOG. A consequência do uso do valor *edge* está representado na figura 2. É perceptível que o problema da criação de valores no HOG é diminuído drasticamente.

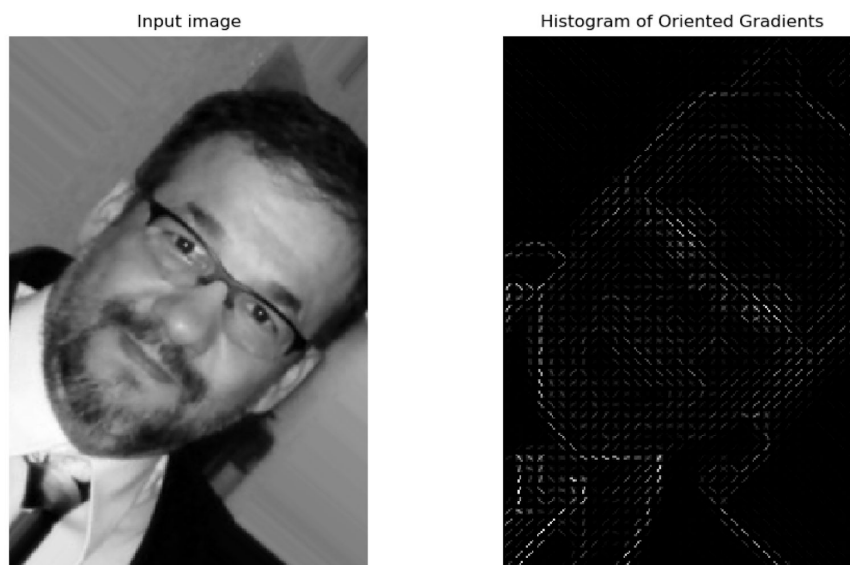


Figura 2 - Preenchimento dos limites da imagem rotacionada com o valor *edge* no atributo *mode*

3. Acurácias dos modelos utilizados

Na execução do projeto, foi variado os atributos dos classificadores KNN e MLP de diversas formas. Para o KNN, foi analisado os valores para $k = 3$, $k = 5$ e $k = 7$. Para o MLP, foram variados os atributos:

- *Learning rate*, com os valores 0.01, 0.1 e 1;
- *Momentum*, com os valores 0.1, 0.5 e 1;
- Tamanho da primeira camada, com 10, 30, 50, 80 e 100 neurônios;
- Tamanho da segunda camada, com 0 (equivalente à inexistência da segunda camada), 30, 50, 80 e 100 neurônios.

No caso da aplicação do algoritmo KNN, foram obtidas as acurácias para cada k utilizado para cada base de dados, sendo inseridas nos gráficos abaixo representados pelas figuras 3 e 4.

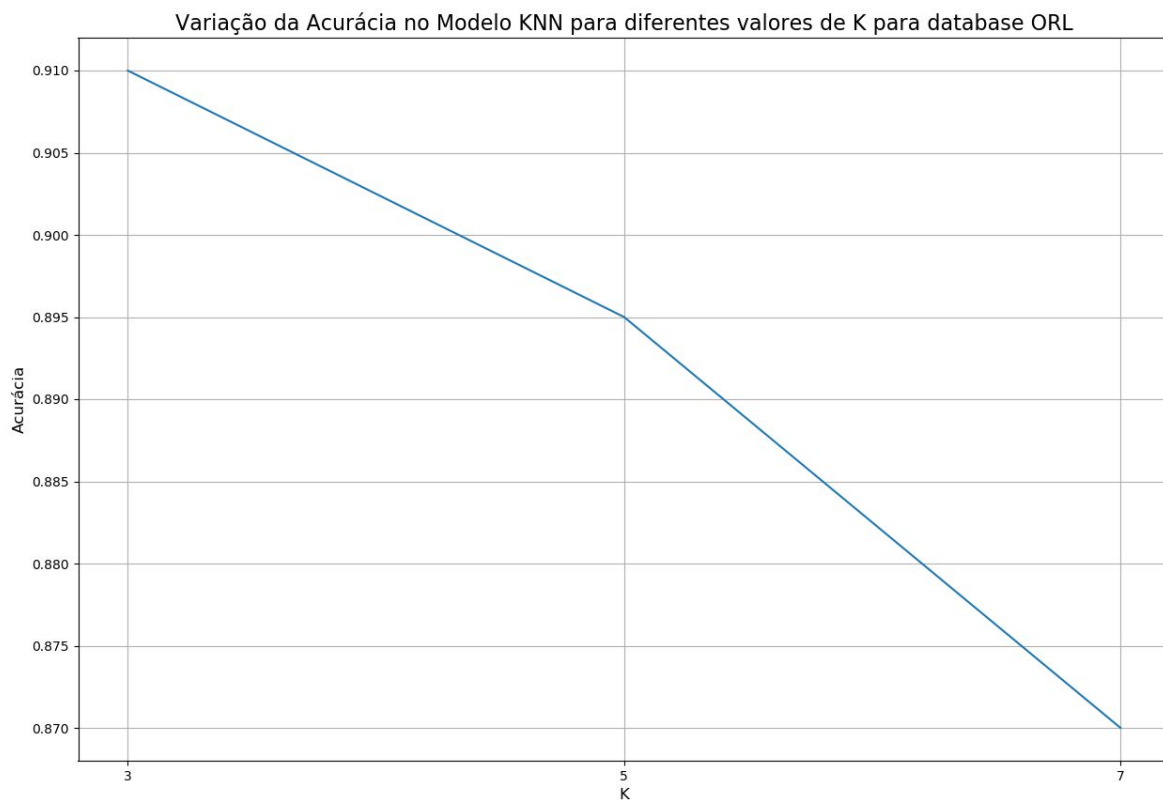


Figura 3 - Acurácias obtidas para o algoritmo KNN com $k = 3$, $k = 5$ e $k = 7$ na base de dados ORL

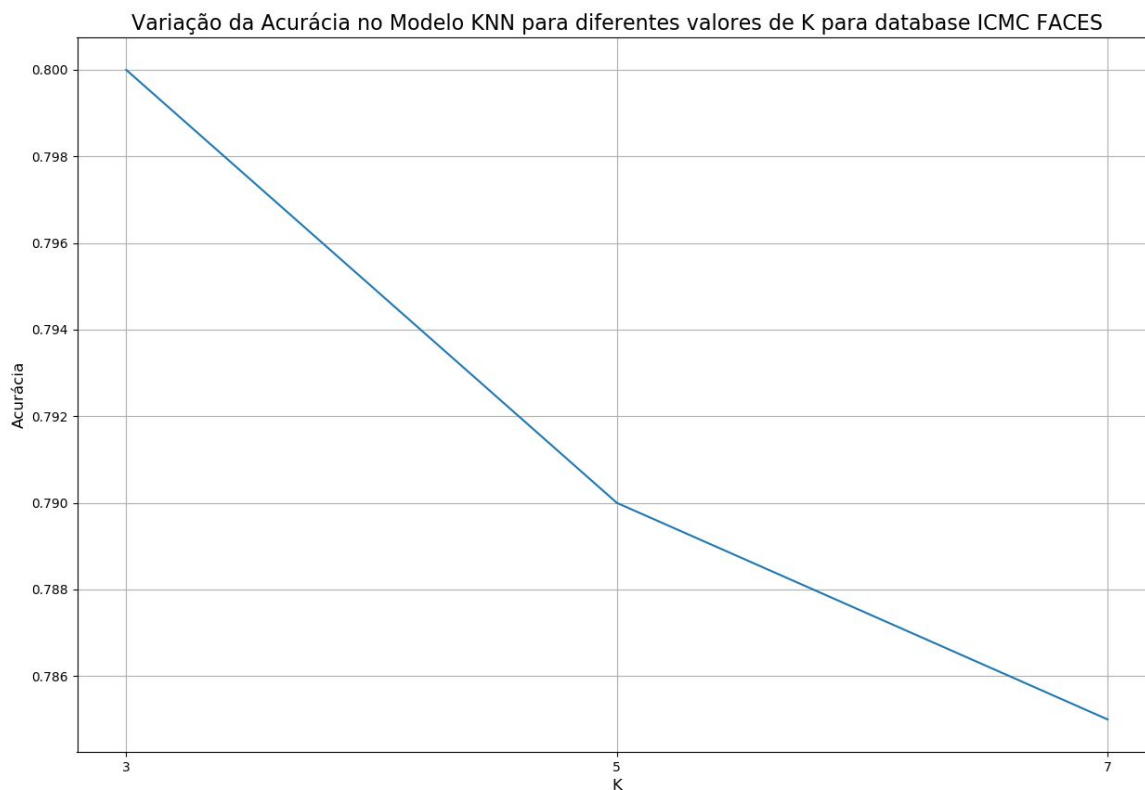


Figura 4 - Acurácias obtidas para o algoritmo KNN com $k = 3$, $k = 5$ e $k = 7$ na base de dados Pessoas ICMC

Para a aplicação do algoritmo MLP, como são muitos casos de teste, foram criados arquivos CSV para guardar as acurácias obtidas para cada teste realizado, com especificação clara de cada atributo utilizado em cada execução do algoritmo. Será deixado como anexo os CSV das acurácias para cada uma das duas bases de dados, com nomes “*mlp_scores_ORL.csv*” e “*mlp_scores_ICMC FACES.csv*”.

4. Resultados para os melhores classificadores

Na execução dos classificadores, obtemos o melhor desempenho:

- Para a base de dados ORL:
 - KNN com $k = 3$, com acurácia de 91%
 - MLP com *learning rate* de 1.0, *momentum* de 1.0, 100 neurônios na primeira camada e sem segunda camada, com acurácia de 96.5%
- Para a base de dados Pessoas ICMC:
 - KNN com $k = 3$, com acurácia de 80%
 - MLP com *learning rate* de 1.0, *momentum* de 1.0, 80 neurônios na primeira camada e sem segunda camada, com acurácia de 86%

As precisões para cada classificador, calculada em cada classe das bases de dados, estão representadas na tabela 1 e 2.

Classe	Precisão (KNN na condição de melhor desempenho)	Precisão (MLP na condição de melhor desempenho)
0	83.33%	100.00%
1	71.43%	100.00%
2	100.00%	100.00%
3	100.00%	90.91%
4	81.82%	100.00%
5	100.00%	90.91%
6	63.64%	100.00%
7	100.00%	90.91%
8	100.00%	100.00%
9	100.00%	100.00%
10	100.00%	100.00%
11	100.00%	100.00%
12	90.00%	90.91%
13	90.91%	90.91%
14	90.91%	90.91%
15	100.00%	100.00%
16	83.33%	90.91%
17	100.00%	100.00%
18	88.89%	100.00%
19	100.00%	100.00%

Tabela 1 - Precisão de cada classe para o algoritmo KNN, com $k = 3$, junto da precisão de cada classe para o algoritmo MLP, com *learning rate* igual a 1.0, *momentum* igual a 1.0, 100 neurônios na primeira camada e sem segunda camada, na base de dados ORL

Classe	Precisão (KNN na condição de melhor desempenho)	Precisão (MLP na condição de melhor desempenho)
0	57.14%	90.00%
1	70.00%	100.00%
2	87.50%	100.00%
3	81.82%	100.00%
4	76.92%	75.00%
5	100.00%	100.00%
6	70.00%	80.00%
7	47.62%	69.23%
8	100.00%	100.00%
9	72.73%	72.73%
10	88.89%	81.82%
11	100.00%	90.00%
12	100.00%	100.00%
13	88.89%	80.00%
14	72.73%	88.89%
15	100.00%	71.43%
16	100.00%	70.00%
17	72.73%	88.89%
18	100.00%	100.00%
19	100.00%	90.00%

Tabela 2 - Precisão de cada classe para o algoritmo KNN, com $k = 3$, junto da precisão de cada classe para o algoritmo MLP, com *learning rate* igual a 1.0, *momentum* igual a 1.0, 80 neurônios na primeira camada e sem segunda camada, na base de dados Pessoas ICMC

As matrizes de confusão para cada classificador, para cada um dos conjuntos de dados, estão representadas pelas figuras 5, 6, 7 e 8.

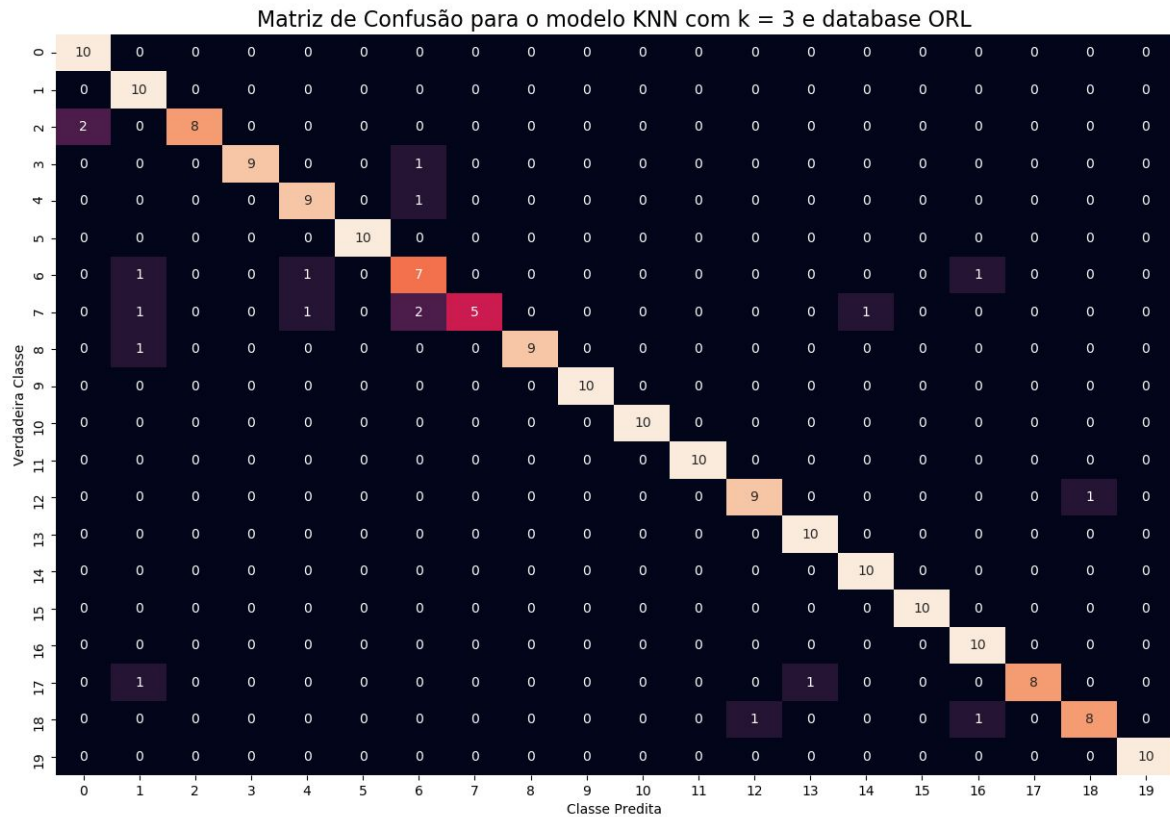


Figura 5 - Matriz de confusão para o classificador KNN com $k = 3$, para o conjunto de dados ORL

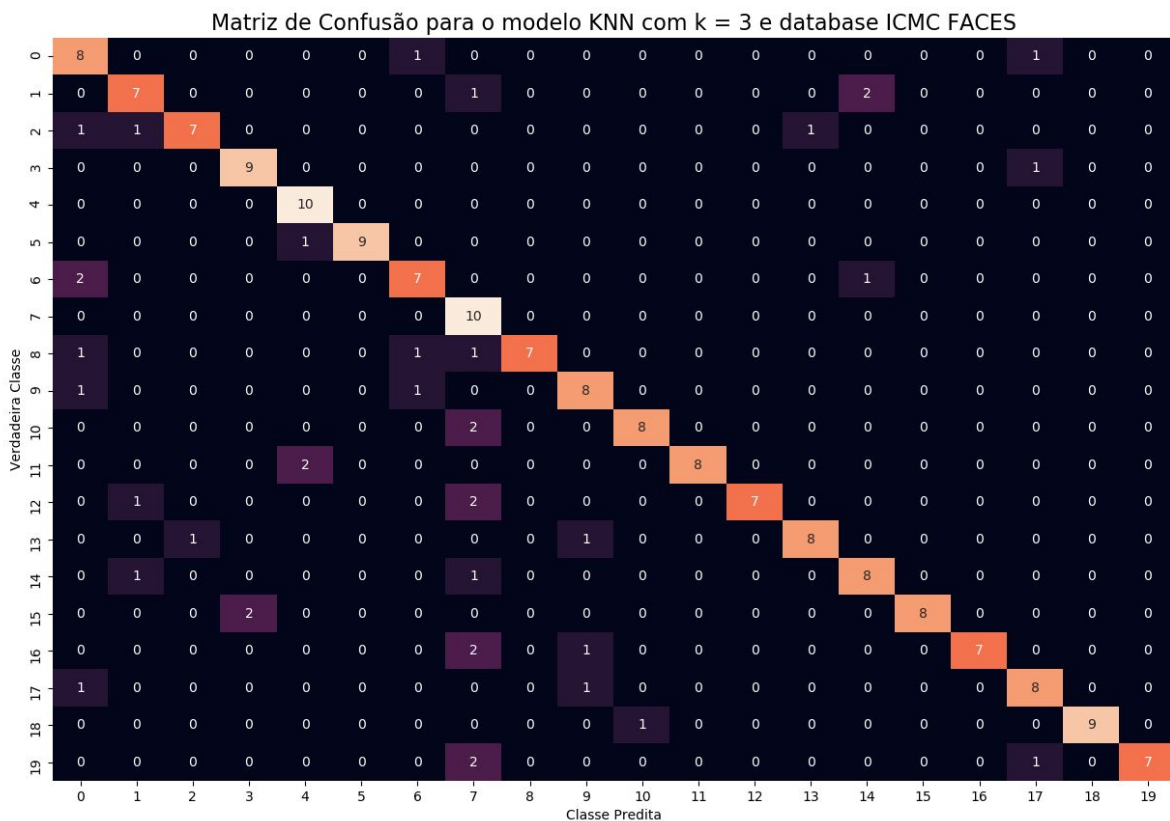


Figura 6 - Matriz de confusão para o classificador KNN com $k = 3$, para o conjunto de dados Pessoas ICMC

Matriz de Confusão para o modelo MLP: Learning Rate: 1.0000, Momentum: 1.00, Layer 1 Size: 100, Layer 2 Size: 0 e database ORI

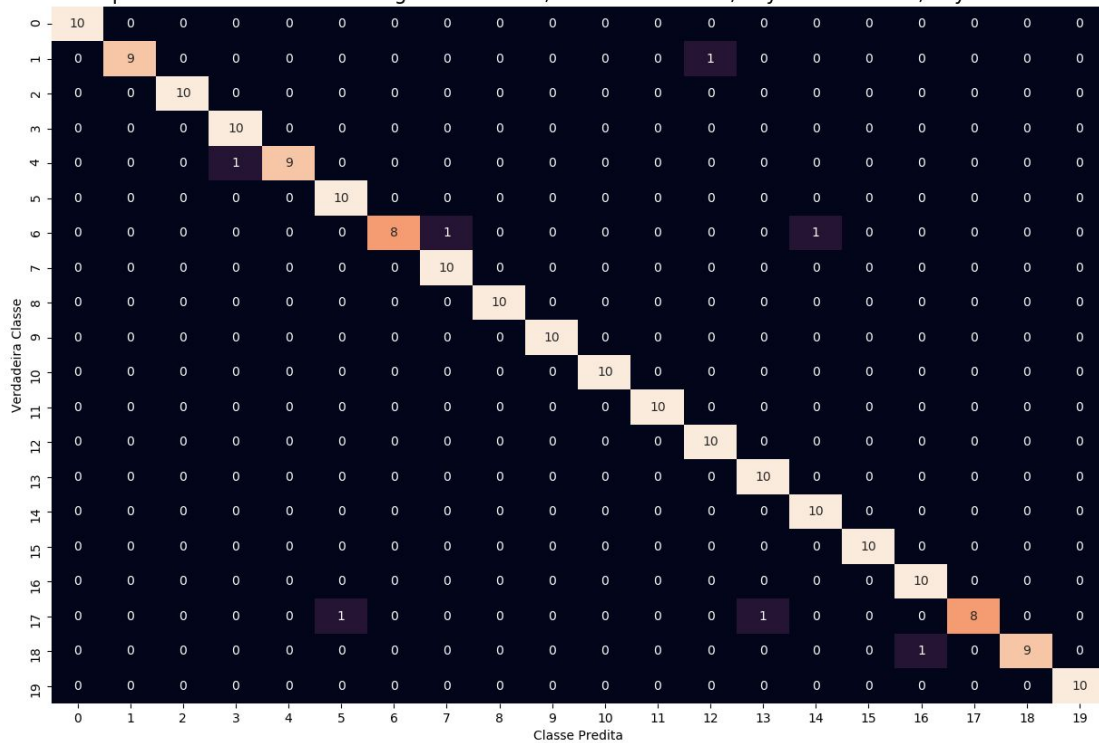


Figura 7 - Matriz de confusão para o classificador MLP, com *learning rate* igual a 1.0, *momentum* igual a 1.0, 100 neurônios na primeira camada e sem segunda camada, para o conjunto de dados ORI

Matriz de Confusão para o modelo MLP: Learning Rate: 1.0000, Momentum: 1.00, Layer 1 Size: 80, Layer 2 Size: 0 e database ICMC FA

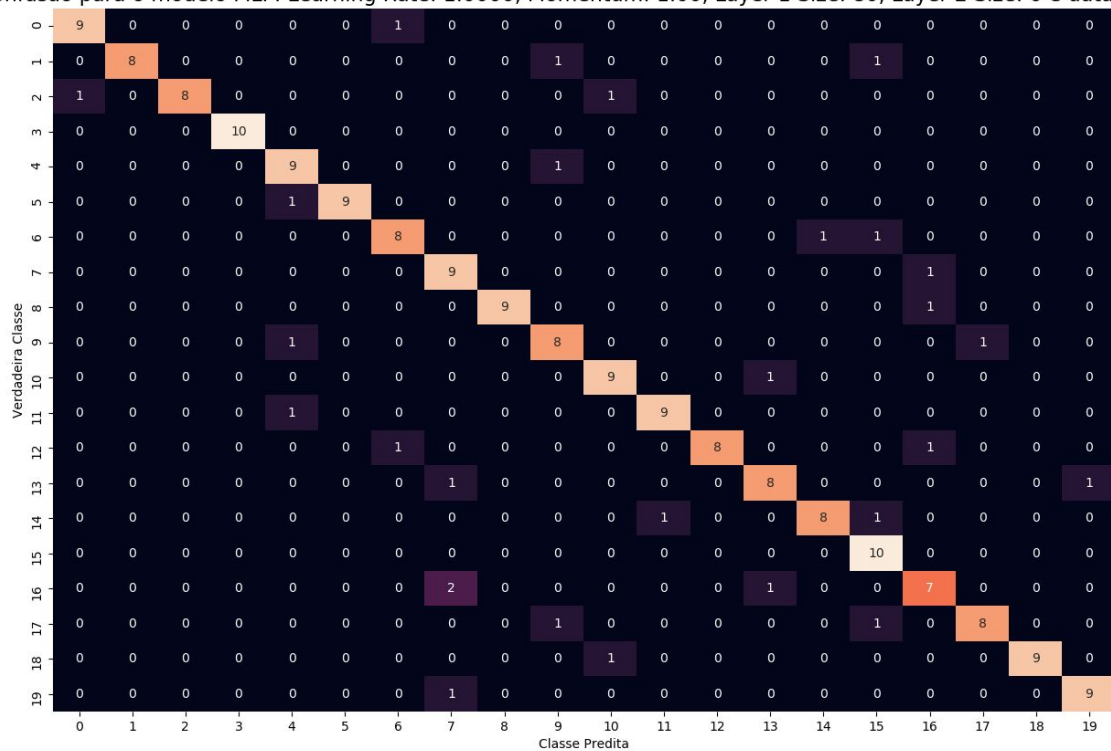


Figura 8 - Matriz de confusão para o classificador MLP, com *learning rate* igual a 1.0, *momentum* igual a 1.0, 80 neurônios na primeira camada e sem segunda camada, para o conjunto de dados Pessoas ICMC

Observações sobre a base de dados ORL:

É interessante se notar que, para o KNN, a pessoa s7 (classe 6), foi a que apresentou maior dificuldade para ser classificada, sendo seguida por s2 (classe 1). Olhando essas classes na matriz de confusão, percebe-se que outras pessoas foram classificadas como pertencentes a essas classes. Investigando as imagens, percebe-se que s2 tirou todas as fotos com óculos e s7 apresenta pouca variação da posição do rosto de uma foto para outra, o que pode explicar a baixa precisão dessas classes. O classificador MLP, por sua vez, foi capaz de melhor separar melhor as classes apesar dessas características.

Observações sobre a base de dados ICMC:

As pessoas referentes às classes 0 (p1) e 7 (p8) foram as que apresentaram maior dificuldade para classificação utilizando o KNN. Analisando a base de dados com as imagens artificiais, percebe-se que a pessoa p1 utiliza óculos e o fundo da imagem não tem muita informação em comparação às imagens dos demais professores com óculos. p8, por sua vez, possui um fundo de foto com quase nenhum elemento e não há muito contraste entre o fundo da imagem e o cabelo da pessoa. O modelo MLP foi capaz de manter a precisão de todas as classes acima de 70% com exceção para p8. Olhando a matriz de confusão para o MLP, observa-se que a classe 16 (p17) foi a que apresentou menos exemplo corretamente classificados, sendo alguns considerados p8. A presença do óculos na imagem de p17 pode ser uma explicação para a dificuldade de classificação.

5. Resultados para os melhores classificadores com o uso do PCA

Ao utilizar o PCA nas bases de dados e utilizando os melhores classificadores elencados anteriormente, obtivemos as acurácias representadas pela tabela 3.

Base de dados	Variância recuperada dos dados originais	Classificador (na melhor configuração)	Acurácia com dados originais	Acurácia com PCA
ORL	51.07%	KNN	91%	83.5%
		MLP	96.5%	80%
Pessoas ICMC	52.09%	KNN	80%	71%
		MLP	86%	69%

Tabela 3 - Acurácia de cada classificador de melhor desempenho após o uso do método PCA

As matrizes de confusão para cada classificador após a aplicação do PCA, para cada um dos conjuntos de dados, estão representadas pelas figuras 9, 10, 11 e 12.

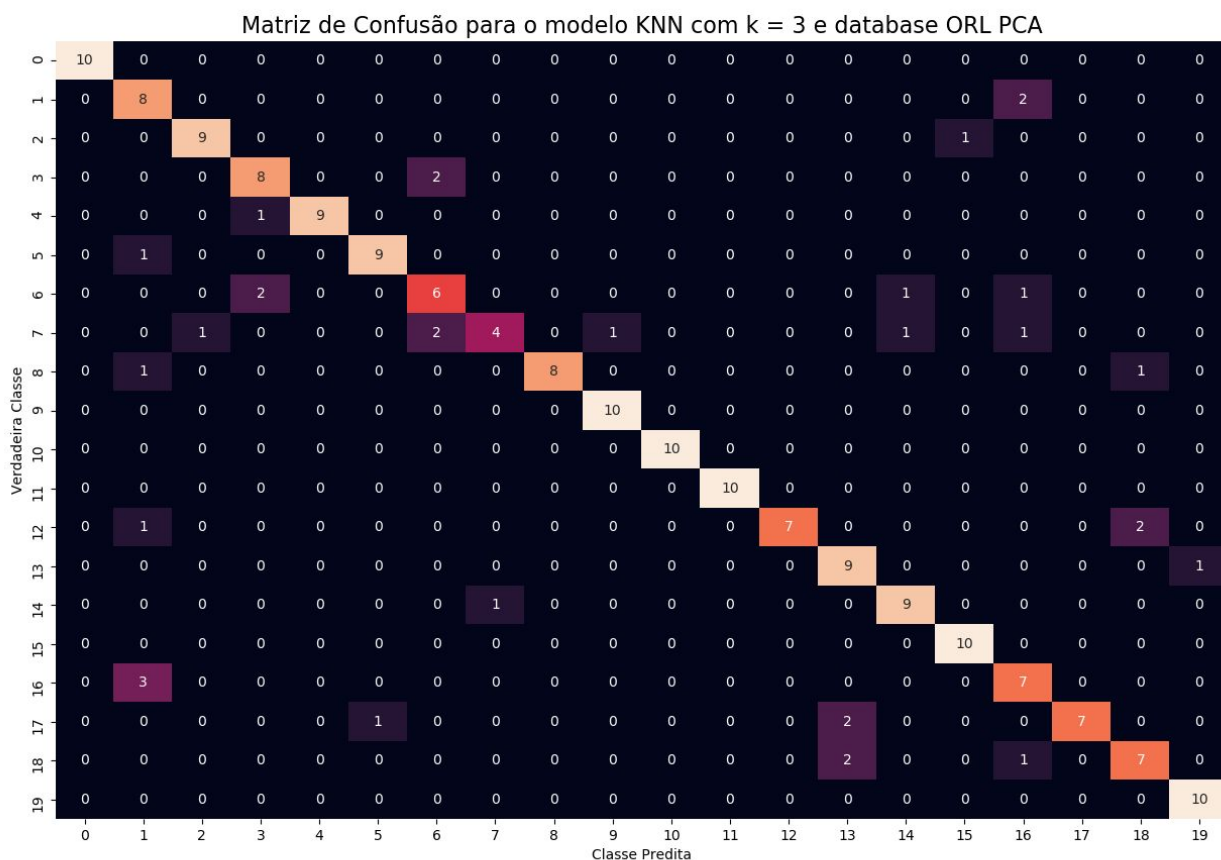


Figura 9 - Matriz de confusão para o classificador KNN com $k = 3$ após aplicar PCA, para o conjunto de dados ORL

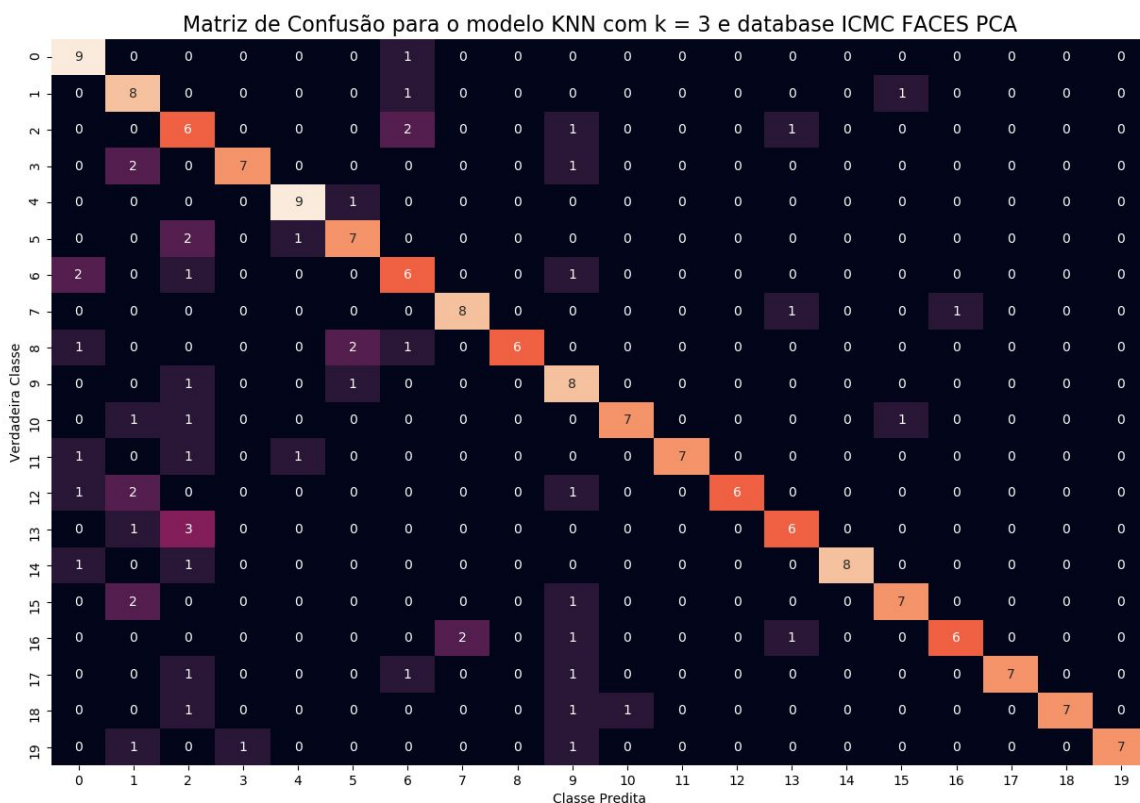


Figura 10 - Matriz de confusão para o classificador KNN com $k = 3$ após aplicar PCA, para o conjunto de dados Pessoas ICMC

Matriz de Confusão para o modelo MLP: Learning Rate: 1.0000, Momentum: 1.00, Layer 1 Size: 100, Layer 2 Size: 0 e database ORL P

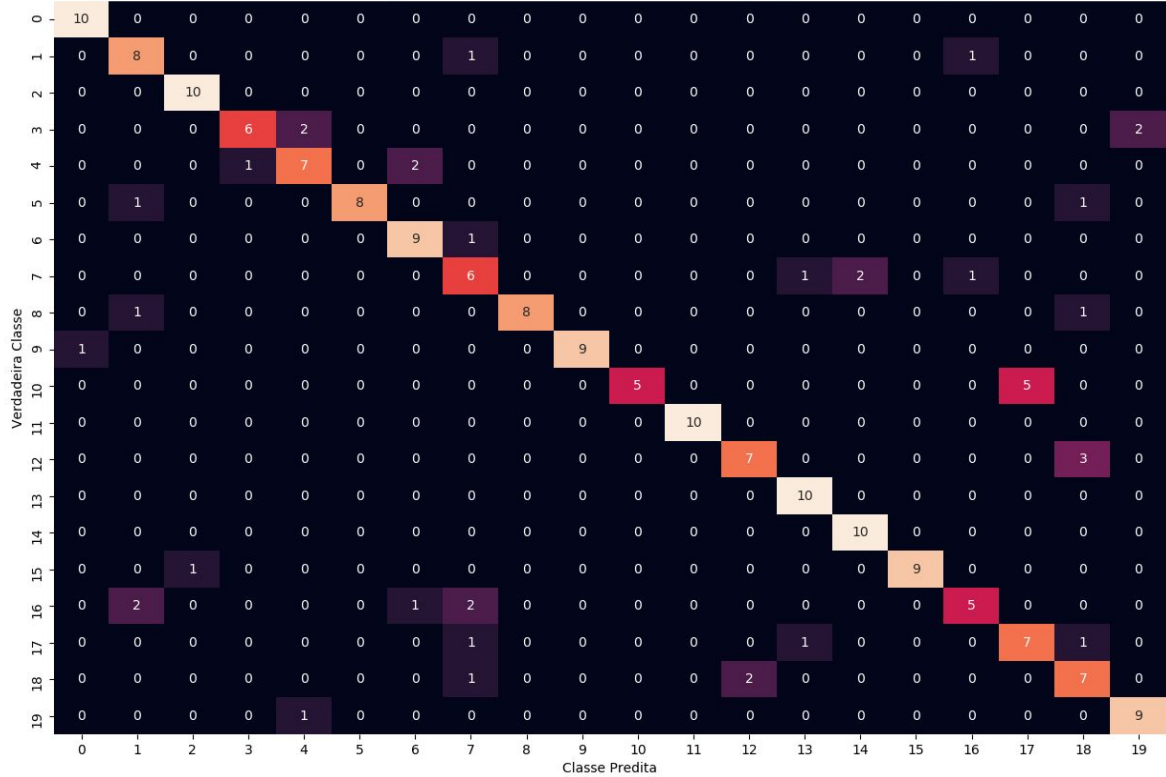


Figura 11 - Matriz de confusão para o classificador MLP, com *learning rate* igual a 1.0, *momentum* igual a 1.0, 100 neurônios na primeira camada e sem segunda camada, após aplicar PCA, para o conjunto de dados ORL

Matriz de Confusão para o modelo MLP: Learning Rate: 1.0000, Momentum: 1.00, Layer 1 Size: 80, Layer 2 Size: 0 e database ICMC FACE

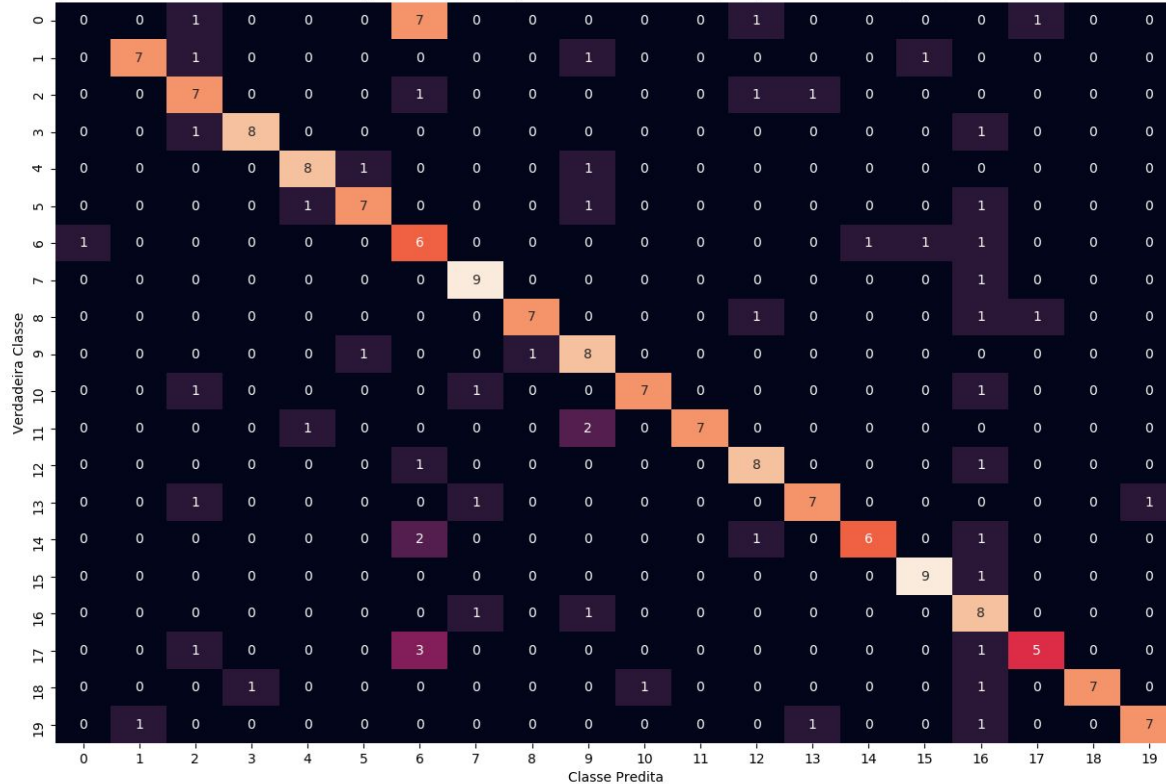


Figura 12 - Matriz de confusão para o classificador MLP, com *learning rate* igual a 1.0, *momentum* igual a 1.0, 80 neurônios na primeira camada e sem segunda camada, após aplicar PCA, para o conjunto de dados Pessoas ICMC

A precisão de cada classe para cada classificador, após o uso do PCA, está representada nas tabelas 4 e 5, abaixo.

Classe	Precisão (KNN na condição de melhor desempenho) com dados originais	Precisão (KNN na condição de melhor desempenho) com PCA	Precisão (MLP na condição de melhor desempenho) com dados originais	Precisão (MLP na condição de melhor desempenho) com PCA
0	83.33%	100.00%	100.00%	90.91%
1	71.43%	57.14%	100.00%	66.67%
2	100.00%	90.00%	100.00%	90.91%
3	100.00%	72.73%	90.91%	85.71%
4	81.82%	100.00%	100.00%	70.00%
5	100.00%	90.00%	90.91%	100.00%
6	63.64%	60.00%	100.00%	75.00%
7	100.00%	80.00%	90.91%	50.00%
8	100.00%	100.00%	100.00%	100.00%
9	100.00%	90.91%	100.00%	100.00%
10	100.00%	100.00%	100.00%	100.00%
11	100.00%	100.00%	100.00%	100.00%
12	90.00%	100.00%	90.91%	77.78%
13	90.91%	69.23%	90.91%	83.33%
14	90.91%	81.82%	90.91%	83.33%
15	100.00%	90.91%	100.00%	100.00%
16	83.33%	58.33%	90.91%	71.43%
17	100.00%	100.00%	100.00%	58.33%
18	88.89%	70.00%	100.00%	53.85%
19	100.00%	90.91%	100.00%	81.82%

Tabela 4 - Precisão de cada classe para o algoritmo KNN, com $k = 3$, junto da precisão de cada classe para o algoritmo MLP, com *learning rate* igual a 1.0, *momentum* igual a 1.0, 100 neurônios na primeira camada e sem segunda camada, após a aplicação do PCA na base de dados ORL

Classe	Precisão (KNN na condição de melhor desempenho) com dados originais	Precisão (KNN na condição de melhor desempenho) com PCA	Precisão (MLP na condição de melhor desempenho) com dados originais	Precisão (MLP na condição de melhor desempenho) com PCA
0	57.14%	60.00%	90.00%	0.00%
1	70.00%	47.06%	100.00%	87.50%
2	87.50%	33.33%	100.00%	53.85%
3	81.82%	87.50%	100.00%	88.89%
4	76.92%	81.82%	75.00%	80.00%
5	100.00%	63.64%	100.00%	77.78%
6	70.00%	50.00%	80.00%	30.00%
7	47.62%	80.00%	69.23%	75.00%
8	100.00%	100.00%	100.00%	87.50%
9	72.73%	47.06%	72.73%	57.14%
10	88.89%	87.50%	81.82%	87.50%
11	100.00%	100.00%	90.00%	100.00%
12	100.00%	100.00%	100.00%	66.67%
13	88.89%	66.67%	80.00%	77.78%
14	72.73%	100.00%	88.89%	85.71%
15	100.00%	77.78%	71.43%	81.82%
16	100.00%	85.71%	70.00%	40.00%
17	72.73%	100%	88.89%	71.43%
18	100.00%	100%	100.00%	100.00%
19	100.00%	100%	90.00%	87.50%

Tabela 5 - Precisão de cada classe para o algoritmo KNN, com $k = 3$, junto da precisão de cada classe para o algoritmo MLP, com *learning rate* igual a 1.0, *momentum* igual a 1.0, 80 neurônios na primeira camada e sem segunda camada, após a aplicação do PCA na base de dados Pessoas ICMC

Após a aplicação do PCA, observa-se que houve uma diminuição na acurácia obtida pelos modelos KNN e MLP. De forma geral, as precisões do MLP foram mais afetadas por essa transformação. Conclui-se, portanto, que a redução da dimensão dos dados retirou informações que permitiam uma separação mais clara entre as classes consideradas.

6. Exemplo de Execução do Código

```
===== PRE PROCESSING =====
Lendo dados ORL Faces
ORL FACES:
    Número de pessoas: 20
    Faces por pessoas: [10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,
10]
Pessoas ICMC:
    Número de pessoas: 20
    Faces por pessoas: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
--- Gerando faces extras para base Pessoas ICMC ---
Pessoas ICMC - Extendido:
    Número de pessoas: 20
    Faces por pessoas: [10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,
10]
Salvando Pessoas ICMC - Extendido
===== HOG EXTRACTION =====
Gerando novas HOG features
===== TRAINING SECTION =====
Dimensão Features Originais:
    ORL: 1536
    ICMC FACES: 1280

=====
DATABASE: ORL
=====
===== KNN =====
Training with k = 3
Accuracy for k = 3: 0.9099999999999999
Training with k = 5
Accuracy for k = 5: 0.8949999999999999
Training with k = 7
Accuracy for k = 7: 0.8700000000000001
Melhor valor para k: 3 com acurácia 91.00%
Database: ORL, Modelo: KNN com k = 3
    Acurácia: 0.91000
    Precisão:
        Classe 0: 83.33%
        Classe 1: 71.43%
        Classe 2: 100.00%
        Classe 3: 100.00%
        Classe 4: 81.82%
        Classe 5: 100.00%
```

Classe 6: 63.64%
Classe 7: 100.00%
Classe 8: 100.00%
Classe 9: 100.00%
Classe 10: 100.00%
Classe 11: 100.00%
Classe 12: 90.00%
Classe 13: 90.91%
Classe 14: 90.91%
Classe 15: 100.00%
Classe 16: 83.33%
Classe 17: 100.00%
Classe 18: 88.89%
Classe 19: 100.00%

===== MLP =====

training model: learning_rate: 0.0100, momentum: 0.10, Layer 1 Size: 10, Layer 2 Size: 0

...

training model: learning_rate: 1.0000, momentum: 1.00, Layer 1 Size: 100, Layer 2 Size: 100

Melhor configuração para MLP

Melhor learning_rate: 1.0

Melhor momentum: 1.0

Melhor layer 1 size: 100

Melhor layer 2 size: 0

Database: ORL, Modelo: MLP: Learning Rate: 1.0000, Momentum: 1.00, Layer 1 Size: 100, Layer 2 Size: 0

Acurácia: 0.96500

Precisão:

Classe 0: 100.00%
Classe 1: 100.00%
Classe 2: 100.00%
Classe 3: 90.91%
Classe 4: 100.00%
Classe 5: 90.91%
Classe 6: 100.00%
Classe 7: 90.91%
Classe 8: 100.00%
Classe 9: 100.00%
Classe 10: 100.00%
Classe 11: 100.00%
Classe 12: 90.91%
Classe 13: 90.91%
Classe 14: 90.91%
Classe 15: 100.00%
Classe 16: 90.91%
Classe 17: 100.00%
Classe 18: 100.00%
Classe 19: 100.00%

===== PCA =====

PCA:

Dimensão: 11

Variância recuperada dos dados originais: 51.07%

Database: ORL PCA, Modelo: KNN com k = 3

Acurácia: 0.83500

Precisão:

Classe 0: 100.00%
Classe 1: 57.14%
Classe 2: 90.00%
Classe 3: 72.73%
Classe 4: 100.00%
Classe 5: 90.00%
Classe 6: 60.00%
Classe 7: 80.00%
Classe 8: 100.00%
Classe 9: 90.91%
Classe 10: 100.00%
Classe 11: 100.00%
Classe 12: 100.00%
Classe 13: 69.23%
Classe 14: 81.82%
Classe 15: 90.91%
Classe 16: 58.33%
Classe 17: 100.00%
Classe 18: 70.00%
Classe 19: 90.91%

**Database: ORL PCA, Modelo: MLP: Learning Rate: 1.0000, Momentum: 1.00, Layer 1 Size: 100,
Layer 2 Size: 0**

Acurácia: 0.80000

Precisão:

Classe 0: 90.91%
Classe 1: 66.67%
Classe 2: 90.91%
Classe 3: 85.71%
Classe 4: 70.00%
Classe 5: 100.00%
Classe 6: 75.00%
Classe 7: 50.00%
Classe 8: 100.00%
Classe 9: 100.00%
Classe 10: 100.00%
Classe 11: 100.00%
Classe 12: 77.78%
Classe 13: 83.33%
Classe 14: 83.33%
Classe 15: 100.00%
Classe 16: 71.43%
Classe 17: 58.33%
Classe 18: 53.85%
Classe 19: 81.82%

=====
DATABASE: ICMC FACES
=====

===== KNN =====

Training with k = 3

Accuracy for k = 3: 0.8

Training with k = 5

Accuracy for k = 5: 0.79

Training with k = 7

Accuracy for k = 7: 0.7849999999999999

Melhor valor para k: 3 com acurácia 80.00%

Database: ICMC FACES, Modelo: KNN com k = 3

Acurácia: 0.80000

Precisão:

Classe 0: 57.14%

Classe 1: 70.00%

Classe 2: 87.50%

Classe 3: 81.82%

Classe 4: 76.92%

Classe 5: 100.00%

Classe 6: 70.00%

Classe 7: 47.62%

Classe 8: 100.00%

Classe 9: 72.73%

Classe 10: 88.89%

Classe 11: 100.00%

Classe 12: 100.00%

Classe 13: 88.89%

Classe 14: 72.73%

Classe 15: 100.00%

Classe 16: 100.00%

Classe 17: 72.73%

Classe 18: 100.00%

Classe 19: 100.00%

===== MLP =====

training model: learning_rate: 0.0100, momentum: 0.10, Layer 1 Size: 10, Layer 2 Size: 0

...

training model: learning_rate: 1.0000, momentum: 1.00, Layer 1 Size: 100, Layer 2 Size: 100

Melhor configuração para MLP

Melhor learning_rate: 1.0

Melhor momentum: 1.0

Melhor layer 1 size: 80

Melhor layer 2 size: 0

Database: ICMC FACES, Modelo: MLP: Learning Rate: 1.0000, Momentum: 1.00, Layer 1 Size: 80, Layer 2 Size: 0

Acurácia: 0.86000

Precisão:

Classe 0: 90.00%

Classe 1: 100.00%

Classe 2: 100.00%

Classe 3: 100.00%

Classe 4: 75.00%

Classe 5: 100.00%

Classe 6: 80.00%

Classe 7: 69.23%
Classe 8: 100.00%
Classe 9: 72.73%
Classe 10: 81.82%
Classe 11: 90.00%
Classe 12: 100.00%
Classe 13: 80.00%
Classe 14: 88.89%
Classe 15: 71.43%
Classe 16: 70.00%
Classe 17: 88.89%
Classe 18: 100.00%
Classe 19: 90.00%

===== PCA =====

PCA:

Dimensão: 11

Variância recuperada dos dados originais: 52.09%

Database: ICMC FACES PCA, Modelo: KNN com k = 3

Acurácia: 0.71000

Precisão:

Classe 0: 60.00%
Classe 1: 47.06%
Classe 2: 33.33%
Classe 3: 87.50%
Classe 4: 81.82%
Classe 5: 63.64%
Classe 6: 50.00%
Classe 7: 80.00%
Classe 8: 100.00%
Classe 9: 47.06%
Classe 10: 87.50%
Classe 11: 100.00%
Classe 12: 100.00%
Classe 13: 66.67%
Classe 14: 100.00%
Classe 15: 77.78%
Classe 16: 85.71%
Classe 17: 100.00%
Classe 18: 100.00%
Classe 19: 100.00%

Database: ICMC FACES PCA, Modelo: MLP: Learning Rate: 1.0000, Momentum: 1.00, Layer 1

Size: 80, Layer 2 Size: 0

Acurácia: 0.69000

Precisão:

Classe 0: 0.00%
Classe 1: 87.50%
Classe 2: 53.85%
Classe 3: 88.89%
Classe 4: 80.00%
Classe 5: 77.78%
Classe 6: 30.00%

Classe 7: 75.00%
Classe 8: 87.50%
Classe 9: 57.14%
Classe 10: 87.50%
Classe 11: 100.00%
Classe 12: 66.67%
Classe 13: 77.78%
Classe 14: 85.71%
Classe 15: 81.82%
Classe 16: 40.00%
Classe 17: 71.43%
Classe 18: 100.00%
Classe 19: 87.50%