

Identificação tipológica e similaridade entre monumentos religiosos de Portugal: Abordagem com redes neurais convolucionais e redes siamesas

Trabalho de Grupo realizado no âmbito da Unidade Curricular de Aprendizagem Profunda para Visão por Computador do 1º ano do Mestrado em Ciência de Dados

Diogo Freitas, 104841, MCD-LCD-A1

Diogo_Alexandre_Freitas@iscte-iul.pt

João Francisco Botas, 104782, MCD-LCD-A1

Joao_Botas@iscte-iul.pt

Miguel Gonçalves, 105944, MCD-LCD-A1

Miguel_Goncalves_Pereira@iscte-iul.pt

Ricardo Galvão, 105285, MCD-LCD-A1

Araujo_Galvao@iscte-iul.pt

Índice

Resumo	Página 1
1. Introdução	Página 1
2. Metodologia e descrição geral do sistema	Página 1
2.1. CNNs para classificação multi-classe	Página 2
2.2. Redes Siamesas	Página 3
3. Dados	Página 5
4. Resultados	Página 7
4.1. Modelo multi-classe por tipologia de monumentos - Problema (1)	Página 8
4.2. Modelo multi-classe em monumentos religiosos - Problema (2)	Página 9
4.3. Redes siamesas - Problema (3)	Página 10
5. Conclusões	Página 12
Bibliografia	Página 13
6. Anexos	Página 14
Anexo A - Arquitetura redes	Página 14
Rede para o Problema (1)	Página 14
Rede para o Problema (2)	Página 15
Anexo B - Data Augmentation	Página 16
Anexo C - Evolução das métricas e matrizes de confusão	Página 17
Rede para o Problema (1)	Página 17
Rede para o Problema (2)	Página 18
Anexo D - Tentativas falhadas com redes siamesas no decorrer do <i>pipeline</i>	Página 19
Anexo E - Desenvolvimento do projeto - GitHub	Página 21
<i>Streamlit</i> como ideia para demo na apresentação do projeto	Página 21

Resumo

Este trabalho explora a aplicação de Redes Neuronais Convolucionais (CNNs) e redes siamesas para a identificação e comparação tipológica de monumentos portugueses com base em imagens. Foram definidas três problemáticas: (1) classificação multi-classe de monumentos e paisagens em 11 categorias; (2) categorização detalhada de patrimónios religiosos em 5 classes; e (3) medição de similaridade entre imagens por meio de redes siamesas com *Triplet Loss*. Utilizaram-se arquiteturas como EfficientNet e variações personalizadas e aplicadas técnicas de *Data Augmentation*, normalização e seleção de dados representativos através de *clustering*. Os modelos alcançaram bons resultados no problema (1), desempenho moderado no desafiante problema (2) e geraram representações coerentes no problema (3), que foram úteis para tarefas de correspondência visual. Os resultados demonstram o potencial destas abordagens no reconhecimento de patrimónios, mesmo em cenários com elevada semelhança visual entre classes.

1. Introdução

Com o crescimento exponencial na utilização das redes sociais e da partilha massiva de imagens georreferenciadas, tornou-se particularmente pertinente investigar métodos automáticos de identificação e classificação de locais com base nas suas características visuais. Neste contexto, as Redes Neuronais Convolucionais (CNNs) destacam-se como uma solução robusta para problemas de reconhecimento de *features* e classificação de imagens.

A escolha do domínio de estudo - monumentos localizados em território português (do Continente e Ilhas) - é motivada pela sua relevância cultural, histórica e turística. Portugal é reconhecido internacionalmente como um destino de excelência, como o demonstram as múltiplas distinções atribuídas pelo [World Travel Awards](#), nos últimos anos. Assim, explorar soluções de inteligência artificial aplicadas ao reconhecimento visual do património nacional, reveste-se de grande interesse prático, académico e social.

O presente trabalho tem como objetivo desenvolver e avaliar modelos de visão computacional, com base em CNNs e variações destas, capazes de classificar imagens de monumentos portugueses, segundo a sua tipologia. Através de abordagens supervisionadas e de redes siamesas, pretende-se investigar a eficácia destes modelos na diferenciação visual entre categorias patrimoniais distintas, bem como explorar estratégias de representação e comparação entre imagens.

2. Metodologia e descrição geral do sistema

Durante o desenvolvimento do projeto, foram definidas três problemáticas principais, centradas em monumentos e paisagens do território português, que são organizadas segundo um grau crescente de complexidade ao longo do estudo. Ao longo do texto, estas serão referidas como problema (1), (2) e (3), respetivamente, e são descritas da seguinte forma:

- (1) Classificação automática de imagens com monumentos e paisagens como patrimónios religiosos, palácios, castelos, quintas, torres, entre outros. As imagens das *landmarks* são agrupadas em 11 categorias

distintas¹, que configuram um problema de classificação multi-classe. Este primeiro desafio pretende fornecer uma visão preliminar sobre a capacidade dos modelos distinguirem diferentes tipologias de monumentos e paisagens, que frequentemente podem apresentar algumas características visuais semelhantes;

- (2) Aprofundamento do problema anterior com um foco nos patrimónios religiosos, ao subdividi-los em cinco categorias distintas: Igrejas, Basílicas, Capelas, Catedrais e Santuários; é também um problema de classificação multi-classe. A decisão das cinco classes é guiada, sobretudo, pelo tipo de estruturas à disposição e a algumas fontes procuradas via web². Este segundo problema surge como um desdobramento natural do (1), mas motivado pela presença expressiva de património religioso em Portugal. Para além do interesse temático, esta tarefa apresenta-se como um teste rigoroso à capacidade discriminativa das CNNs, dada a possível semelhança visual e arquitetónica entre as classes;
- (3) Envolve a utilização de redes neuronais siamesas para medir a similaridade entre pares de imagens, com o objetivo de relacionar monumentos e paisagens visuais semelhantes. Esta metodologia revela-se útil em aplicações de rotulagem automática de fotografias, onde não existe a necessidade da localização explícita, e jogos interativos como o *GeoGuessr*, em que a estimativa da localização baseia-se apenas em dados visuais.

2.1. CNNs para classificação multi-classe

Para a construção das redes utilizadas nos Problemas (1) e (2) recorrem-se a arquiteturas convolucionais profundas, com abordagens distintas para cada caso.

No Problema (1) recorre-se à utilização de uma rede pré treinada, com base na arquitetura do *EfficientNetB0*. Esta escolha é sustentada por:

- (i) a arquitetura apresentar uma maior profundidade em termos de capacidade representacional e ser especialmente adequada para classificação com um número elevado de classes, ao permitir a aprendizagem de padrões visuais mais complexos e discriminativos;
- (ii) ser reconhecida na literatura por ser mais eficiente em termos de desempenho e utilização de recursos computacionais (parâmetros) face a arquiteturas tradicionais [1];
- (iii) permitir testar, numa fase inicial, a eficácia da transferência de conhecimento, explorando pesos previamente treinados, com as camadas convolucionais congeladas e apenas as camadas densas adaptadas ao contexto específico de classificação da tipologia de monumentos portugueses.

Já para o Problema (2), é proposta uma arquitetura *custom*, desenvolvida de raiz. As camadas desta rede são organizadas em três blocos principais, diferenciados principalmente pela configuração dos filtros e das camadas que realizam o *max pooling*. Cada bloco aplica convoluções com tamanhos de *kernel* 3x3, seguidas de normalização e ativação, sendo que no final de cada bloco é realizada uma operação de *max pooling* para reduzir a dimensionalidade espacial das representações e tentar extrair as características mais notórias e salientes. Explica-se a utilidade de algumas destas camadas da rede (ver Anexo A para arquitetura completa):

¹Esta divisão é feita segundo a [seguinte fonte](#).

²A separação é proposta segundo a divisão tipológica de patrimónios da [Wikipedia](#). Originalmente constam mais classes, mas só são utilizadas 11 destas, devido à pouca quantidade de imagens de ‘Cruzeiros’ e ‘Quintas classificadas’ e inexistência de ‘Marcos de milha e cruzamento’.

- **Camadas Conv2:** Aplicação sucessiva de convoluções 2D com profundidade crescente ao longo dos três blocos. Inicia-se com 32 filtros, passando para 64 e finalmente 128 (todos de tamanho 3x3) nos três blocos, respetivamente;
- **Batch Normalization:** Aplicado após as camadas convolucionais para estabilizar e acelerar a convergência do treino e mitigar o risco de *overfitting*. É seguido de uma função de ativação ReLu para a introdução de não linearidade;
- **Dropout:** Inserido após as convoluções finais para evitar *overfitting* e promover regularização. Adota-se uma estratégia que, à medida que a rede aumenta, a percentagem de neurónios a serem desligados aumenta gradativamente;
- **GlobalAveragePooling2D e Dense:** Substitui o “Flatten”, ou seja, reduz a informação espacial de cada mapa de características, onde depois tem-se uma camada densa que está totalmente ligada para a classificação final. Neste caso é variável consoante o número de classes do modelo.

Durante o treino das redes, foi utilizado o otimizador Adam, com taxa de aprendizagem inicial de 0,001 e função de perda “sparse_categorical_crossentropy”³, ideal para problemas multi-classe. As redes foram treinadas por 50 epochs com o apoio das seguintes callbacks:

- **F1-ScoreTracker:** Serve para calcular e registar a média ponderada do F1-Score sobre os conjuntos de treino e validação ao final de cada época. Assim permite uma avaliação mais robusta em cenários de classificação multi-classe e com classes potencialmente desequilibradas;
- **ModelCheckpoint:** Guarda os pesos do modelo com o maior valor de F1-Score no conjunto de validação;
- **EarlyStopping:** Interrompe o treino quando o F1-Score deixa de melhorar por 5 épocas consecutivas;
- **ReduceLROnPlateau:** Reduz o *learning rate* em 80%, com um fator de 0,2, sempre que o F1-Score do conjunto de validação estagnar durante três épocas consecutivas, com limite mínimo de 1×10^{-6} .

Ainda, para este conjunto de redes, bem como para as redes siamesas, é aplicado um processo de *Data Augmentation* durante o treino, de forma a aumentar a robustez dos modelos e reduzir o sobreajuste dos modelos. Este processo inclui transformações geométricas - como espelhamento horizontal, *zoom in/out* - e cromáticas - com alterações nos níveis de saturação, brilho e contraste (ver Anexo B para detalhes técnicos).

2.2. Redes Siamesas

Nesta abordagem, utilizam-se duas bibliotecas: TensorFlow e PyTorch. O TensorFlow implementa redes siamesas com pares de imagens, com uma função de perda “Binary Crossentropy”, adequada para distinguir pares semelhantes e diferentes. Contudo, limita-se numa implementação flexível da *Triplet Loss*. Para ultrapassar essa limitação, recorre-se ao PyTorch, que permite um maior controlo no treino e facilita a definição de modelos baseados em triplos de imagens (âncora, positiva e negativa), com *Triplet Loss* [2]. Assim, escolhe-se cada biblioteca conforme as necessidades técnicas da metodologia siamesa a utilizar.

A primeira abordagem implementada recorre ao TensorFlow para desenvolver uma rede siamesa supervisionada, treinada com pares de imagens rotuladas. Cada par é constituído por duas imagens (x_1, x_2) e um rótulo $y \in \{0, 1\}$, sendo que $y = 1$ indica que ambas pertencem à mesma classe (par positivo) e $y = 0$

³Mais sensível à representação das *labels* por inteiros, no índice da classe. Não é necessário fazer *one-hot encoding* (em 1 e 0).

indica classes diferentes (par negativo). O objetivo do modelo é gerar representações vetoriais semelhantes para imagens da mesma classe e dissemelhantes para imagens de classes distintas.

A arquitetura da rede utiliza a EfficientNetB0 pré-treinada no ImageNet como extrator de características, gerando um *embedding* $f(x)$ para cada imagem. Dado um par de imagens, os respectivos *embeddings* são combinados através da concatenação dos próprios vetores, da sua diferença absoluta e da sua multiplicação elemento a elemento:

$$z = [f(x_1), f(x_2), |f(x_1) - f(x_2)|, f(x_1) \cdot f(x_2)] \quad (1)$$

- $f(x_1), f(x_2)$ são as representações (embeddings) das duas entradas;
- $|f(x_1) - f(x_2)|$ é o vetor da diferença absoluta entre os *embeddings*;
- $f(x_1) \cdot f(x_2)$ representa a multiplicação elemento a elemento (produto Hadamard) entre os *embeddings*.

Este vetor combinado é passado por uma camada densa com ativação sigmóide, produzindo uma probabilidade \hat{y} de que o par pertença à mesma classe. A função de perda utilizada é a entropia cruzada binária [3]:

$$L_{\text{BCE}} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (2)$$

- $y \in \{0, 1\}$ é o rótulo verdadeiro da amostra (0 ou 1);
- $\hat{y} \in [0, 1]$ é a probabilidade prevista pelo modelo para a classe 1.

O objetivo é minimizar esta função. O que faz com que o modelo atribua probabilidades altas para a classe correta, aproximando \hat{y} de y . O método de avaliação deste modelo é através das métricas utilizadas num problema de *machine learning* supervisionado de classificação, como a *Accuracy*, *F1_score*, entre outras.

Na segunda abordagem, implementamos o modelo utilizando a *framework* de PyTorch, que nos oferece flexibilidade e controlo detalhado sobre as CNNs. O método baseia-se na aprendizagem por *triplets*, onde o objetivo principal é gerar representações (*embeddings*) para cada imagem que respeitem a relação de similaridade e dissimilaridade entre exemplos. Para isso, selecionamos conjuntos de três imagens: a imagem âncora α , a imagem positiva p que é semelhante à âncora, e a imagem negativa (n), que é diferente. A Figura 1 dá um breve contexto visual do algoritmo:

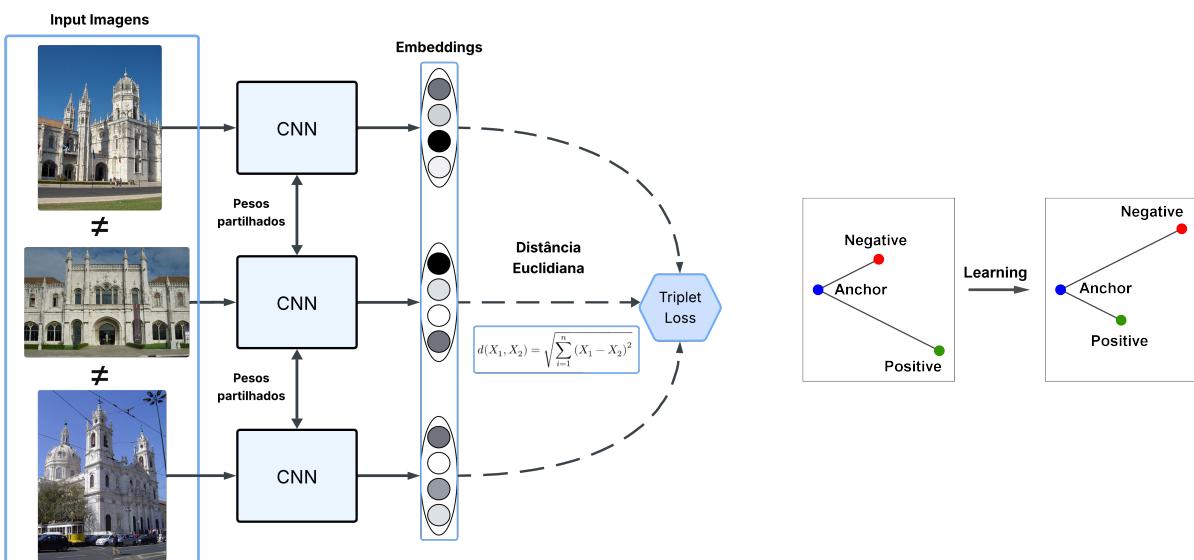


Figura 1: Arquitetura da rede siamesa com *Triplet Loss* (diagrama à direita da figura retirado da Wikipedia)

A função de perda utilizada é a **Triplet Margin Loss**, definida pela seguinte expressão:

$$L(\alpha, p, n) = \max(0, \|f(a) - f(p)\|_2^2 - \|f(a) - f(n)\|_2^2 + \text{margin}) \quad (3)$$

onde:

- $f(x)$ representa o vetor de características (embedding) extraído pelo modelo para a imagem x .
- $\|\cdot\|_2$ é a norma euclidiana.
- $\text{margin} > 0$ é um *hiperparâmetro* que define a diferença mínima desejada entre as distâncias.

A distância Euclidiana [4] entre dois *embeddings* $f(x)$ e $f(y)$, com dimensão m , calcula-se como:

$$d(x, y) = \|f(x) - f(y)\|_2 = \sqrt{\sum_{i=1}^m (f_i(x) - f_i(y))^2} \quad (4)$$

Na prática, a *loss* usa o quadrado da distância Euclidiana para simplificar o cálculo do gradiente, logo:

$$d^2(x, y) = \sum_{i=1}^m (f_i(x) - f_i(y))^2 \quad (5)$$

Esta função de perda força o modelo a aprender um espaço de *embeddings* onde exemplos da mesma classe estejam próximos, enquanto exemplos de classes diferentes se mantenham afastados por pelo menos a margem definida.

Para garantir um treino eficaz, implementámos também uma estratégia de seleção de triplets (**Triplet Sampling Strategy**). Nem todos os triplets contribuem para o gradiente, pois muitos deles geram perda nula — já satisfazendo a margem — enquanto outros são demasiado difíceis, podendo desestabilizar o treino. Adotámos a estratégia de semi-hard negative mining, que seleciona negativos n que satisfazem a condição:

$$d(a, p)^2 < d(a, n)^2 < d(a, p)^2 + \text{margin} \quad (6)$$

Ou seja, os negativos são mais próximos da âncora do que o positivo, mas ainda violam a margem, tornando-os exemplos úteis para o modelo aprender a separar melhor as classes. Quando não existem negativos que cumpram esta condição, escolhemos o negativo mais próximo de a que viole a margem.

Durante o treino, utilizo o otimizador Adam, que ajusta iterativamente os pesos da rede de forma adaptativa para minimizar a função de perda. A rede base é uma EfficientNet-B4 modificada para produzir embeddings de dimensão 128. Esta rede é inicializada com pesos pré-treinados no ImageNet, o que permite reaproveitar representações visuais ricas já aprendidas e ajustá-las ao nosso domínio específico.

3. Dados

A escolha dos dados utilizados foi inspirada no trabalho de T. Weyhand et al. [5], que introduzem o *Google Landmark Dataset v2* (GLDv2) que contém cerca de 5M de imagens de todo o mundo, tendo também sido utilizado como um *dataset* de competição do Kaggle em 2019. Mais detalhes acerca deste conjunto de dados podem ser consultados no [portal oficial](#).

Apesar da abrangência e escala do GLDv2, são identificadas limitações relevantes para o contexto específico de análise de monumentos portugueses. Em particular, os *landmark IDs* são distribuídos por *batches* e não permitem um filtro fácil por país, o que dificulta a identificação direta de locais em Portugal. Para contornar esta limitação, desenvolveu-se um processo de *Web Scraping*, que explora os metadados disponíveis

num ficheiro CSV fornecido no *dataset source*, nomeadamente os *links* associados a cada imagem. A partir destes *links* - apontados para páginas da “[Wikimedia Commons](#)” - é possível recolher automaticamente o país associado a cada imagem e, por conseguinte, fazer um filtro para Portugal especificamente.

Apesar de ser contornado o obstáculo ao obter-se os dados do país, surge um outro onde vários monumentos nacionais, de elevada expressão turística - como o Mosteiro dos Jerónimos - não estão representados no conjunto de dados. Por isso, estes pontos relevantes são retirados a partir da Wikimedia diretamente, através da categoria “[Monumentos Nacionais in Portugal by name](#)”, pelo que garante uma representação mais equilibrada e completa do património português.

Na Figura 2 observa-se o diagrama explicativo desta fase de extração de dados, bem como a visualização de alguns monumentos/paisagens numa amostra de Lisboa.

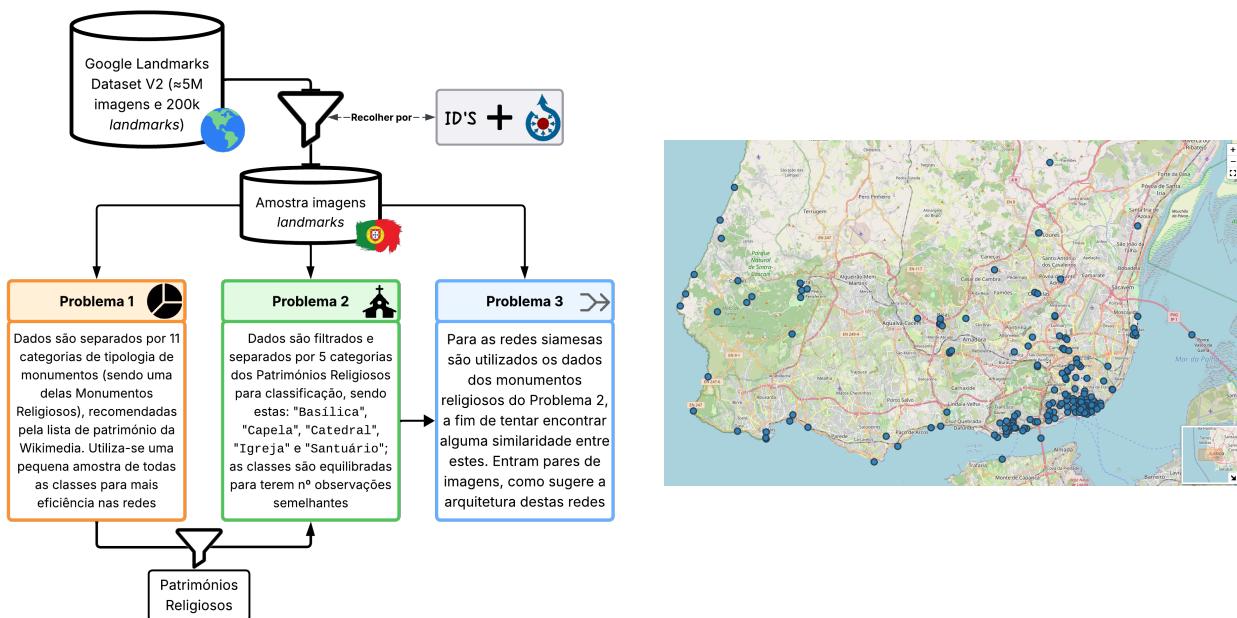


Figura 2: Diagrama da extração de dados e separação por problema (à esquerda) & Densidade de monumentos/paisagens para uma restrição da cidade de Lisboa (à direita)

As imagens recolhidas durante o *pipeline* apresentam dimensões variadas, o que pode comprometer a consistência dos dados e dificultar o treino dos modelos. Para mitigar este problema, todas as imagens são normalizadas para uma resolução de 224×224 pixels, de forma a assegurar a compatibilidade com as arquiteturas utilizadas em (1) e (2).

No processo de construção dos conjuntos de treino e validação para os 2 primeiros problemas, verifica-se a necessidade de “refinar” as classes de interesse, de forma a garantir diversidade e representatividade, sem redundância excessiva. Para isso, recorre-se a uma abordagem de *clustering*, com K-Means, baseada em extração de *features* e hierarquias semânticas, uma técnica inspirada em trabalhos prévios em coleção de imagens [6]. Para cada classe, são selecionadas imagens denominadas como representativas, próximas dos centróides, enquanto imagens redundantes desses mesmos *clusters* são removidas - não representativas (exemplo na Figura 3). Esta estratégia permite reduzir ruído visual, equilibrar as classes a serem utilizadas nas redes e limitar o tamanho total dos dados processados para respeitar restrições de memória RAM.

Inicialmente, com o intuito de restringir o conjunto de dados, foi testada a abordagem de treino de um modelo para distinguir imagens de interior e exterior, com base num [dataset do Kaggle](#). No entanto, os resultados obtidos não demonstraram uma generalização satisfatória e por isso deu-se a esta solução. Esta alternativa disruptiva surgiu como forma preliminar de identificar imagens “outliers” ou mais distantes do padrão geral, permitindo assim uma separação menos rígida e mais exploratória entre imagens de interiores e exteriores de monumentos. Na Figura 3, observa-se que algumas imagens da segunda linha - como (2, 1), (2, 6) e (2, 7) - não apresentam elementos suficientemente marcantes para que um modelo consiga identificar com clareza as características espaciais. Isto deve-se ao facto de não serem, de facto, exteriores - (2, 1) e (2, 2) - ou, então, por apresentarem ruído elevado ou pouco detalhe visual - (2, 6) e (2, 7).



Figura 3: Exemplos de imagens representativas e não representativas em patrimónios religiosos
(imagens na mesma coluna pertencem ao mesmo *cluster*)

Para as redes siamesas considera-se uma proporção de 1/5 das imagens representativas previamente utilizadas no problema 2, patrimónios religiosos. Esta restrição impõe serve para treinar os dados sobre uma amostra significativa, mas limitar a execução para o tempo de treino não ser demasiado extenso e por questões de alocação de memória.

O número de imagens a serem dadas como *input* nos modelos varia de acordo com a problemática, estando representado na Tabela 1 a quantidade de imagens em cada um dos conjuntos na fase de treino.

Tabela 1: Distribuição das imagens utilizadas nas diferentes problemáticas abordadas

Problemática	Distribuição das Imagens	
	Treino	Validação
Classificação 11 classes (1)	3 250	860
Classificação património religioso com 5 classes (2)	2 040	510
Redes siamesas (3)	498	-

É ainda de se reforçar que todo o *pipeline* revela uma grande densidade de monumentos em regiões urbanas. Mesmo numa amostra restrita à cidade de Lisboa, na Figura 2, observa-se uma concentração significativa de imagens, algo que é reflexo da riqueza patrimonial portuguesa, mas que também pode condicionar os tempos de execução e a obtenção de resultados. Esta nuance é tida em conta nesta fase de recolha e preparação de dados para o modelo, bem como na fase de modelação.

4. Resultados

De seguida, apresentam-se os resultados obtidos com os diferentes modelos desenvolvidos e aplicados às problemáticas propostas. Cada modelo é avaliado com base em métricas de desempenho como *Precision*, *Recall* e *F1_Score*, por classe e de forma agregada macro (7), que permite uma análise mais detalhada.

$$F_1^{\text{macro}} = \frac{1}{C} \sum_{i=1}^C F_1^{(i)} = \frac{1}{C} \sum_{i=1}^C 2 \cdot \frac{\text{Precision}^{(i)} \cdot \text{Recall}^{(i)}}{\text{Precision}^{(i)} + \text{Recall}^{(i)}} \quad (7)$$

Sempre que relevante, complementa-se esta discussão de resultados com observações sobre o tempo de treino, classificações incorretas e as matrizes de confusão, com vista a identificar e justificar padrões de erro e notar possíveis sobreposições entre classes semanticamente próximas.

4.1. Modelo multi-classe por tipologia de monumentos - Problema (1)

O modelo para o primeiro problema revela um desempenho global bastante satisfatório, ao apresentar um valor elevado de *accuracy* e uma média macro de *F1-score* equilibrada entre as classes. Este resultado é coerente com a natureza do problema, que combina classes com grande semelhança visual com outras mais facilmente diferenciáveis. Na Tabela 2 encontram-se os valores nas métricas, enquanto a evolução do treino e a matriz de confusão originária podem ser encontradas no Anexo C.

Tabela 2: Métricas - Classificação tipologia 11 classes

Classe	Precision	Recall	F1-Score
Castelos	0,56	0,62	0,59
Castros	0,74	0,81	0,78
Edificações Romanas	0,70	0,61	0,65
Edifícios de Arquitetura Civil	0,62	0,61	0,61
Fortalezas	0,67	0,61	0,64
Monumentos megalíticos	0,74	0,84	0,79
Palácios	0,68	0,57	0,62
Patrimónios religiosos	0,59	0,65	0,62
Pelourinhos	0,70	0,83	0,76
Pontes	0,86	0,80	0,83
Torres	0,83	0,81	0,82
Macr. avg	0,70	0,70	0,70
Accuracy		0,88	

Destaca-se o bom desempenho nas classes ‘Pontes’ e ‘Torres’, que atingem *F1-scores* de 0,83 e 0,82, respetivamente, refletindo a presença de padrões visuais mais consistentes. Em contraste, categorias como ‘Palácios’ ou ‘Edificações Romanas’ registam desempenhos mais modestos que indica uma maior variabilidade interna ou confusão com outras tipologias próximas.

O tempo de treino da rede demorou cerca de 63 minutos com recurso a GPU⁴] que reflete alguma ineficiência da arquitetura utilizada, mesmo perante a diversidade do conjunto de dados. Por fim, é ainda possível notar, através da matriz de confusão, uma tendência à confusão entre “Castelos” e “Fortalezas”, o que é expectável dada a sobreposição histórica e visual entre estas duas categorias – salientando a complexidade semântica envolvida na rotulagem deste tipo de património.

⁴GPU utilizada para a execução dos modelos “custom”: GeForce RTX™ 4060 WINDFORCE OC 8G. Para mais detalhes da implementação de GPU com TensorFlow ver [README do projeto](#).

4.2. Modelo multi-classe em monumentos religiosos - Problema (2)

Os resultados obtidos no segundo modelo não apresentam uma boa performance, como evidenciado pelas métricas da Tabela 3. Estes resultados eram expectáveis, tendo em conta a complexidade do problema: apesar de existir uma diferenciação ligeira entre as estruturas, as imagens globalmente são muito semelhantes, o que dificulta a capacidade do modelo de captar detalhes para uma distinção efetiva.

Tabela 3: Métricas - Classificação património religioso em 5 classes

Classe	Precision	Recall	F1-Score
Basílica	0,38	0,07	0,12
Capela	0,45	0,60	0,51
Catedral	0,55	0,52	0,54
Igreja	0,37	0,30	0,33
Santuário	0,44	0,54	0,48
Macr. avg	0,44	0,41	0,40
Accuracy		0,45	

A classe ‘Basílica’ destaca-se por ser aquela para a qual o modelo apresenta menor capacidade de reconhecimento, comprovado pelos valores significativamente inferiores de *recall* e *F1-score*, o que indica dificuldade do modelo em identificar corretamente exemplos desta classe e em evitar falsos positivos. É possível observar na Figura 4 que as outras classes - maioritariamente a ‘Capela’ - são frequentemente classificadas como ‘Basílica’, o que pode ser justificado pelo número reduzido de imagens provenientes desta categoria. O modelo tem dificuldades claras de generalizar para os dados de testes, tal como se observa pelos valores razoavelmente baixos da *Accuracy* e medidas macro.

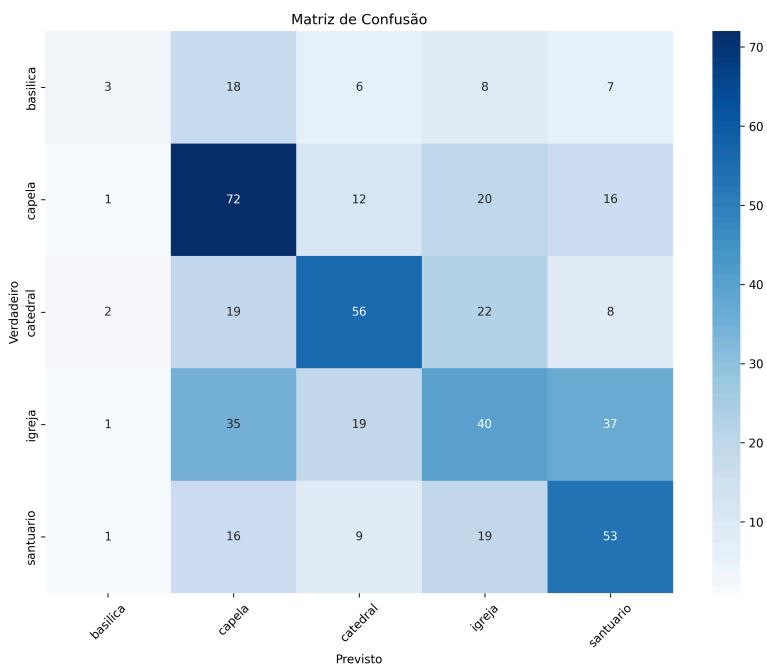


Figura 4: Matriz de confusão - Classificação património religioso em 5 classes

O tempo de treino da rede foi de aproximadamente 8 minutos, um valor significativamente inferior ao do modelo anterior, devido à menor complexidade desta problemática, com menos classes, e à finalização antecipada do treino, onde não foi necessário recorrer à totalidade das *epochs* previstas.

4.3. Redes siamesas - Problema (3)

A abordagem que utiliza redes siamesas para averiguar a similaridade entre imagens passa pela produção de *embeddings* para as diferentes classes consideradas no problema 2 e posteriormente pelo cálculo de uma medida de distância. (Nesta secção será abordada a rede siamesa em que foi aplicado *Triplet Loss*. Por outro lado, a abordagem baseada em entropia cruzada binária revelou-se mais complexa e obteve resultados menos satisfatórios, pelo que não foi alvo de foco principal. Para mais informações sobre esta última, pode ser consultado o trabalho Anexo D)

Pela análise da Figura 5 considera-se que as componentes (PCA) dos *embeddings* apresentam menor separação de classes e dispersão elevada. Já os *embeddings* gerados, quando visualizados com auxílio de t-SNE, apresentam valores semelhantes entre ‘Santuários’, ‘Capelas’ e ‘Basílicas’, como entre ‘Igrejas’ e ‘Catedrais’, havendo reduzida sobreposição entre estes dois grupos. Contudo, as imagens da classe ‘Igreja’ produzem *embeddings* mais dispersos.

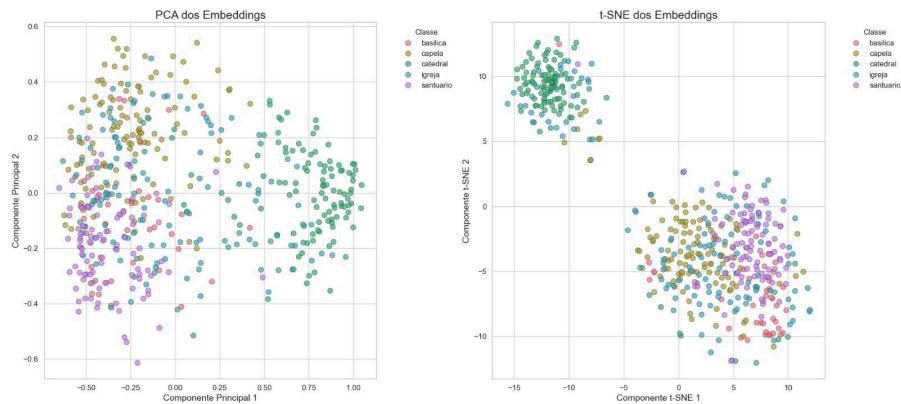


Figura 5: Gráfico de Dispersão dos *Embeddings* de PCA e t-SNE

Através dos *embeddings* gerados foi realizada uma análise intra-classe das anomalias da classe ‘Santuário’. Para esta análise, foi selecionada a imagem mais próxima da média da classe como a mais representativa (à esquerda), em que as imagens restantes são as que se encontram a maior distância desta.

Os resultados foram positivos, sendo que as imagens mais distantes são visivelmente anómalas e não acrescentam valor para a diferenciação entre classes. Observa-se que a 2^a imagem da Figura 6 foi uma exceção que falhou na recolha e tratamento dos dados e apenas apresenta ruído, mas que é facilmente identificado nesta análise.



Figura 6: Anomalias da Classe Santuário

A medida de distância resultante permite averiguar a similaridade entre um par de imagens. Destacam-se imagens das classes ‘Santuário’ e ‘Capela’. Apesar dos *embeddings* se situarem geralmente dentro do mesmo grupo (Figura 5), as distâncias calculadas são superiores a 1, indicando a dissimilaridade entre as duas classes.



Figura 7: Distâncias entre pares de imagens produzidas pela arquitetura de rede siamesa (Santuários vs Capelas)

O tempo de treino desta arquitetura foi de 62,5 minutos, o que evidencia a maior complexidade deste método. Além disso, a utilização de *embeddings* acrescenta um nível adicional de abstração ao problema. Para implementar este modelo, foi necessário restringir a quantidade de imagens utilizadas a um sexto do total, dado que o tempo de treino aumentava quase de forma exponencial. Um número elevado de imagens causava um consumo de memória superior a 21 GB, esgotando os recursos do computador. Foi também necessário ajustar o *batch size* de forma a equilibrar a eficiência com a capacidade computacional disponível.

5. Conclusões

Este trabalho teve como objetivo investigar a aplicação de redes neuronais convolucionais e siamesas na classificação e comparação de imagens de monumentos portugueses, com base em diferentes níveis de granularidade tipológica. Foram formuladas três problemáticas progressivamente mais complexas: uma tarefa inicial de classificação multi-classe com 11 categorias, uma tarefa mais específica centrada em patrimónios religiosos, e uma abordagem de medição de similaridade visual com recurso a redes siamesas e *Triplet Loss*.

Os resultados demonstraram que, com uma preparação cuidadosa dos dados - incluindo normalização, *Data Augmentation* e seleção de amostras representativas - é possível treinar modelos eficazes mesmo em contextos com classes visualmente semelhantes. O uso de modelos pré-treinados, como EfficientNet, mostrou-se vantajoso tanto em desempenho como em eficiência computacional, especialmente nos problemas de classificação. Já as redes siamesas, conjuntamente com a construção de representações vetoriais, revelaram-se úteis para comparação entre imagens, sendo capazes de refletir proximidade semântica exclusivamente com informação visual. No geral, os métodos implementados confirmam o potencial das redes de aprendizagem profunda para aplicações práticas no domínio do reconhecimento patrimonial.

Como trabalho futuro, perspetiva-se o aumento do conjunto de dados e evitar a restrição de dimensão das imagens imposta, ao abranger uma amostra mais representativa dos monumentos existentes em Portugal, ainda que isso implique maior custo computacional e aumento da complexidade do modelo. Para além disso, seria relevante expandir o estudo para outros países ou mesmo considerar cenários “multi-país”, permitindo explorar possíveis variações arquitetónicas entre diferentes contextos geográficos e culturais. Por fim, equaciona-se o desenvolvimento de uma aplicação prática que, a partir de uma imagem, consiga realizar operações conforme as representações extraídas pelas redes convolucionais.

Bibliografia

- [1] M. Tan e Q. V. Le, «EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks», *CoRR*, 2019, doi: [10.48550/arXiv.1905.11946](https://doi.org/10.48550/arXiv.1905.11946).
- [2] F. Ren e S. Xue, «Intention Detection Based on Siamese Neural Network With Triplet Loss», *IEEE Access*, vol. 8, n.º , pp. 82242–82254, 2020, doi: [10.1109/ACCESS.2020.2991484](https://doi.org/10.1109/ACCESS.2020.2991484).
- [3] M. Shorfuzzaman e M. S. Hossain, «MetaCOVID: A Siamese neural network framework with contrastive loss for n-shot diagnosis of COVID-19 patients», *Pattern Recognition*, vol. 113, p. 107700, 2021, doi: <https://doi.org/10.1016/j.patcog.2020.107700>.
- [4] J. Du, W. Fu, Y. Zhang, e Z. Wang, «Advancements in Image Recognition: A Siamese Network Approach». 2024. doi: <https://doi.org/10.56578/ida030202>.
- [5] T. Weyand, A. Araújo, B. Cao, e J. Sim, «Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval», *CoRR*, 2020, doi: [10.48550/arXiv.2004.01804](https://doi.org/10.48550/arXiv.2004.01804).
- [6] Z. Riahi Samani e M. Ebrahimi Moghaddam, «Image Collection Summarization Method Based on Semantic Hierarchies», *AI*, vol. 1, n.º 2, pp. 209–228, 2020, doi: [10.3390/ai1020014](https://doi.org/10.3390/ai1020014).
- [7] A. A. R. Putra e S. Setumin, «The Performance of Siamese Neural Network for Face Recognition using Different Activation Functions», em *2021 International Conference of Technology, Science and Administration (ICTSA)*, 2021, pp. 1–5. doi: [10.1109/ICTSA52017.2021.9406549](https://doi.org/10.1109/ICTSA52017.2021.9406549).

6. Anexos

Anexo A - Arquitetura redes

Rede para o Problema (1)

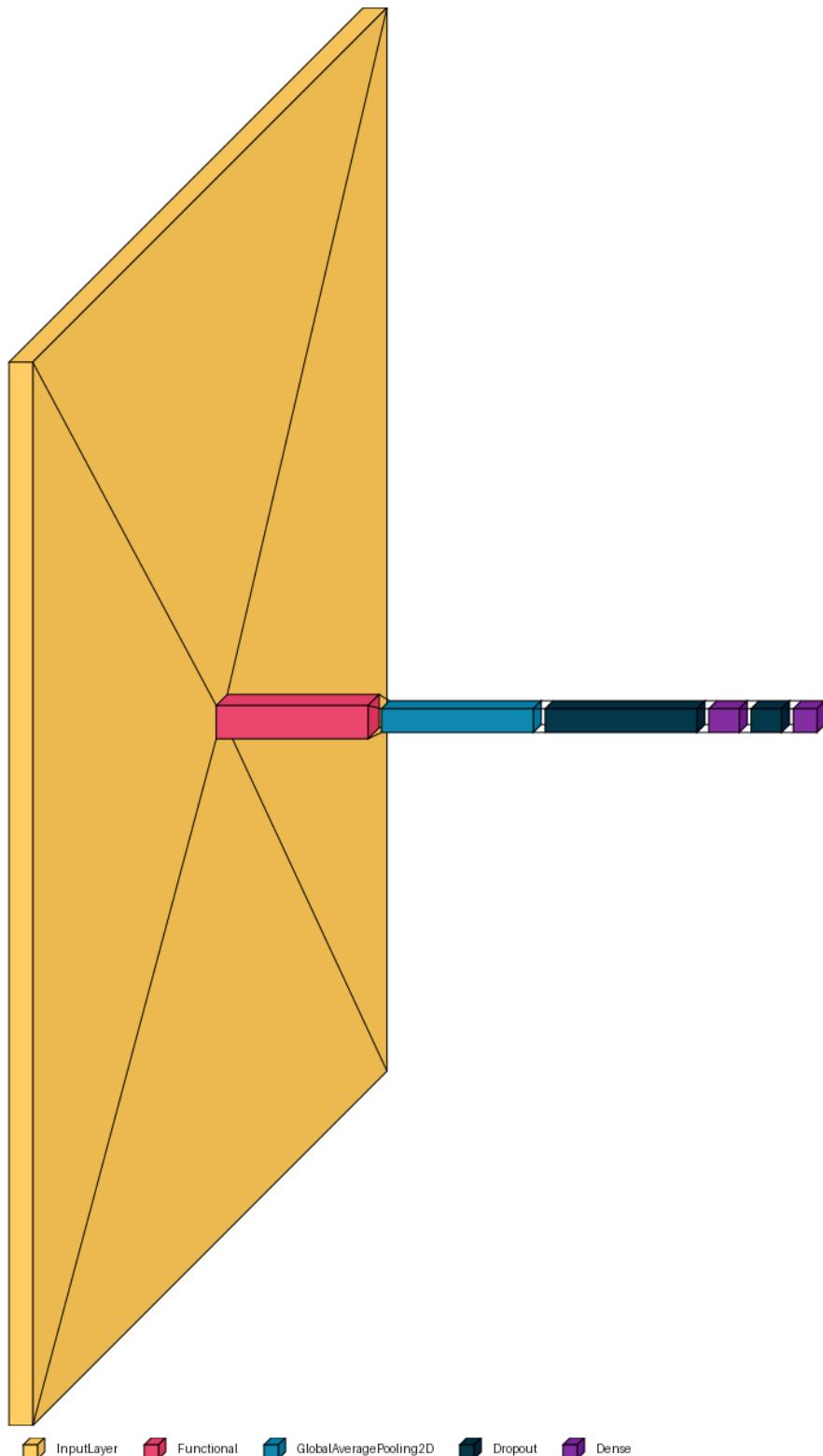


Figura 8: Arquitetura da rede pré-treinada do Problema (1), com as camadas especificadas por cor

Rede para o Problema (2)

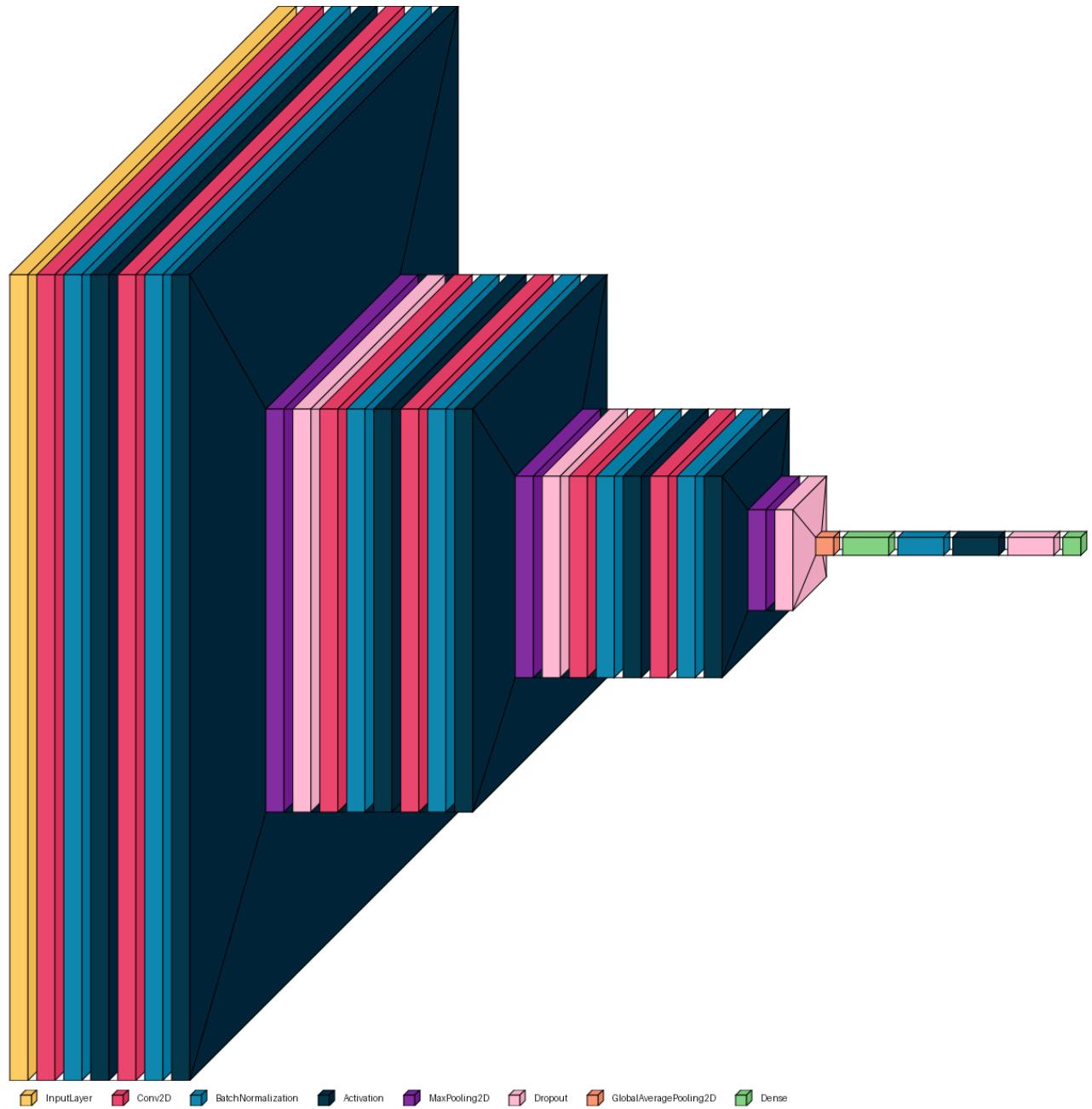


Figura 9: Arquitetura da rede *custom* do Problema (2), com as camadas especificadas por cor

Anexo B - Data Augmentation



Figura 10: Transformação de imagens antes e depois de aplicado *Data Augmentation* - Problema (1)

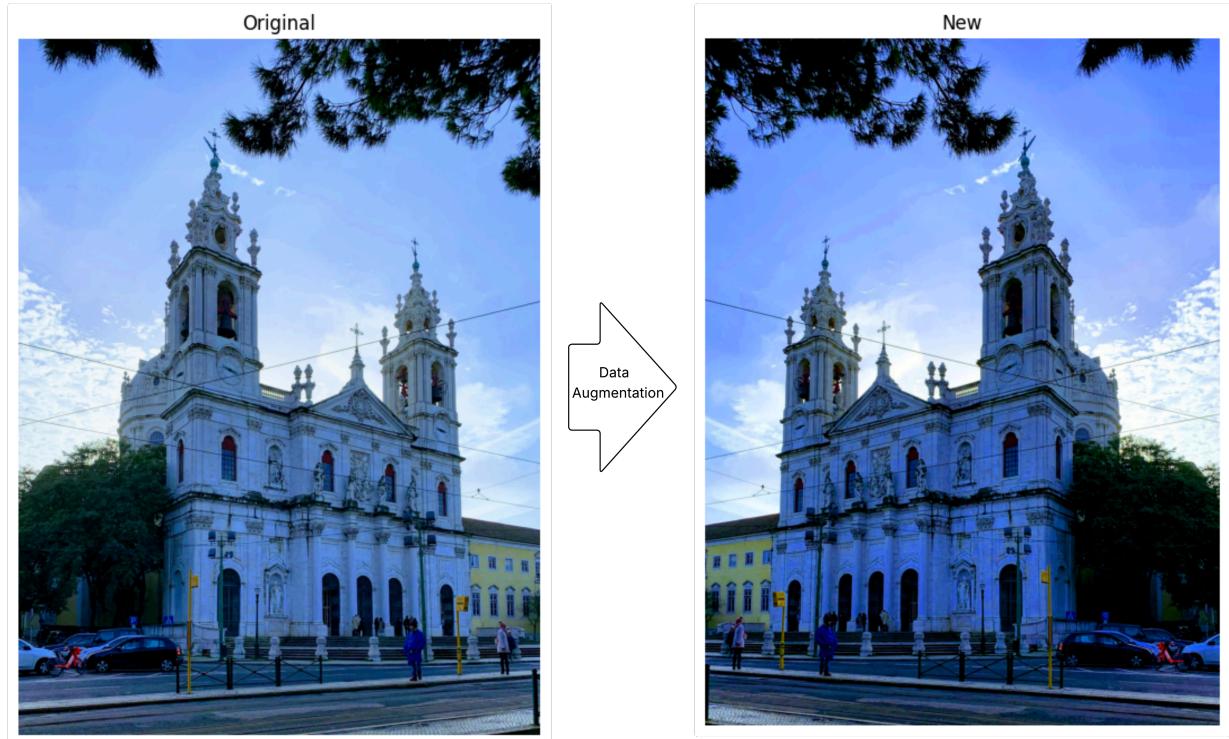


Figura 11: Transformação de imagens antes e depois de aplicado *Data Augmentation* - Problema (2)

Anexo C - Evolução das métricas e matrizes de confusão

Rede para o Problema (1)

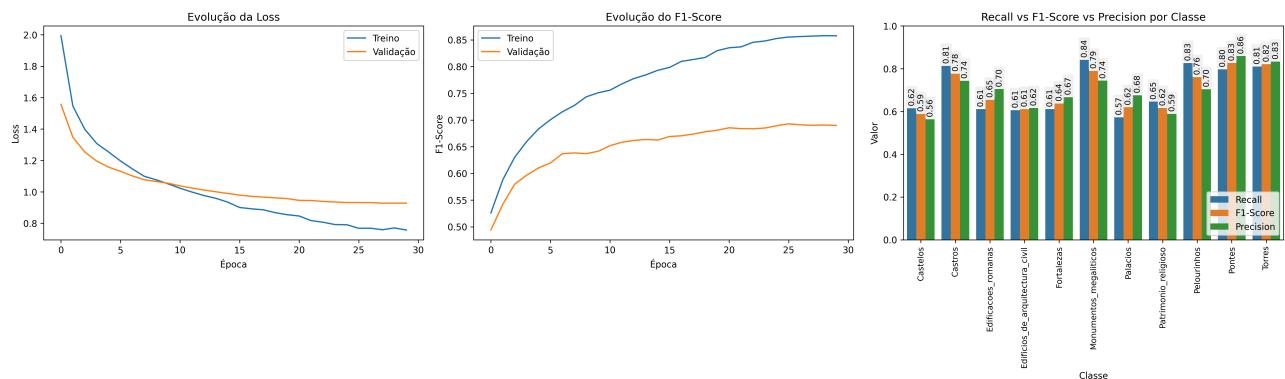


Figura 12: Evolução das métricas - Problema (1)

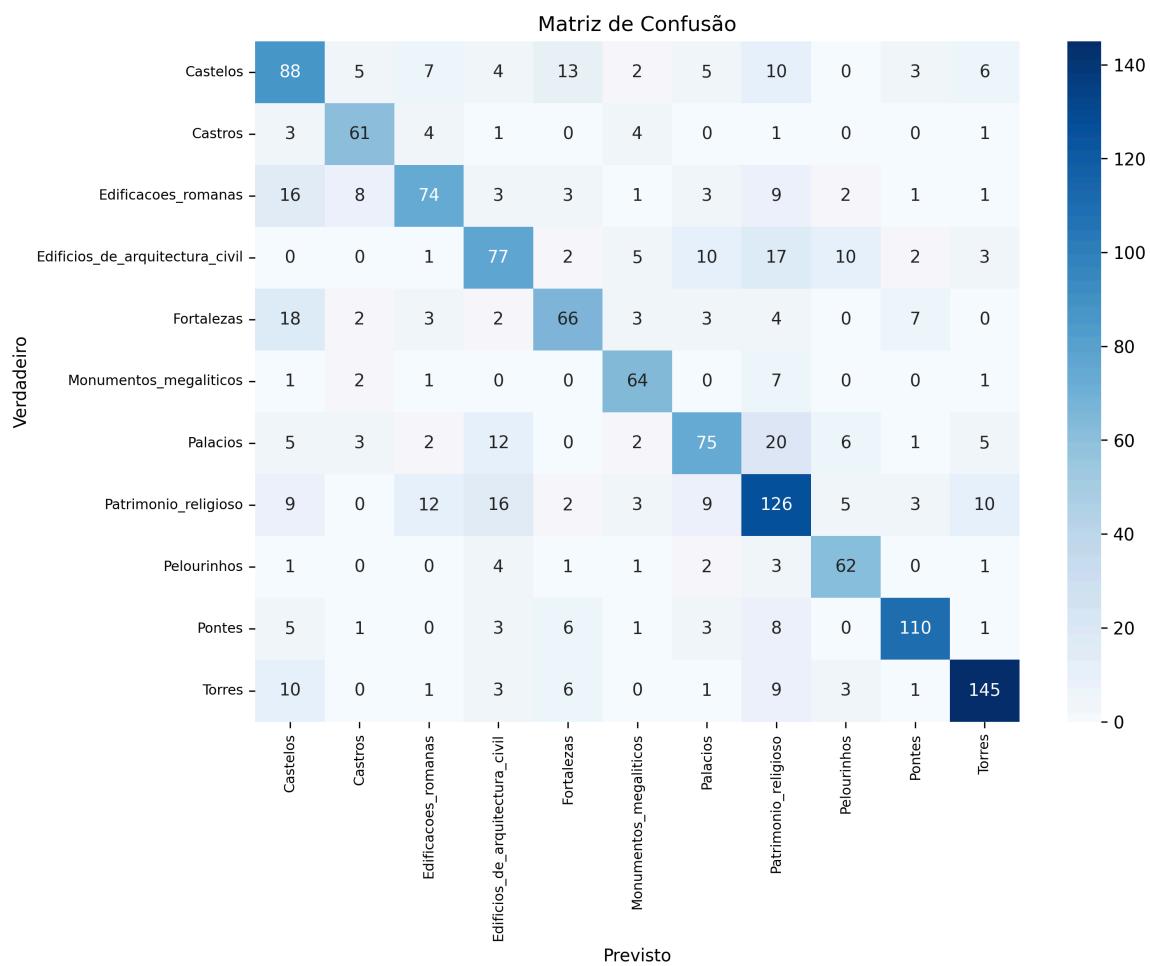


Figura 13: Matriz de confusão - Problema (1)

Rede para o Problema (2)

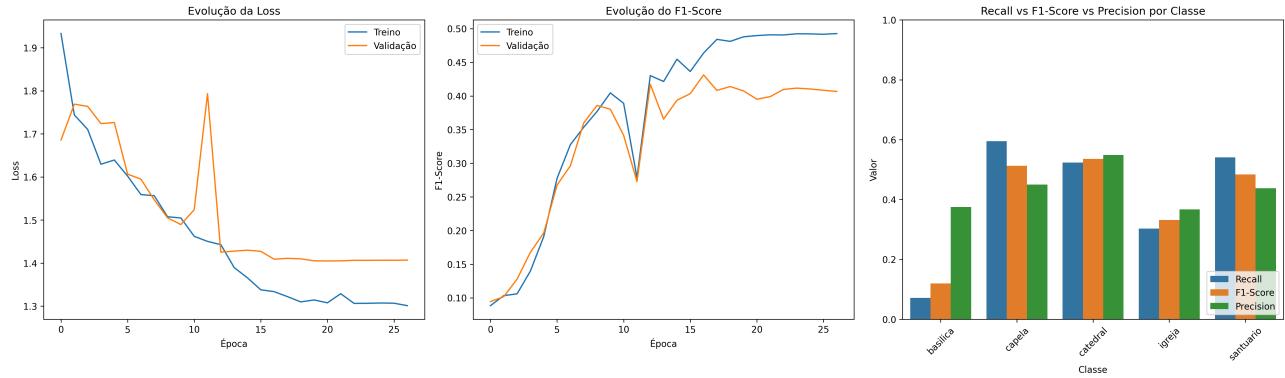


Figura 14: Evolução das métricas - Problema (2)

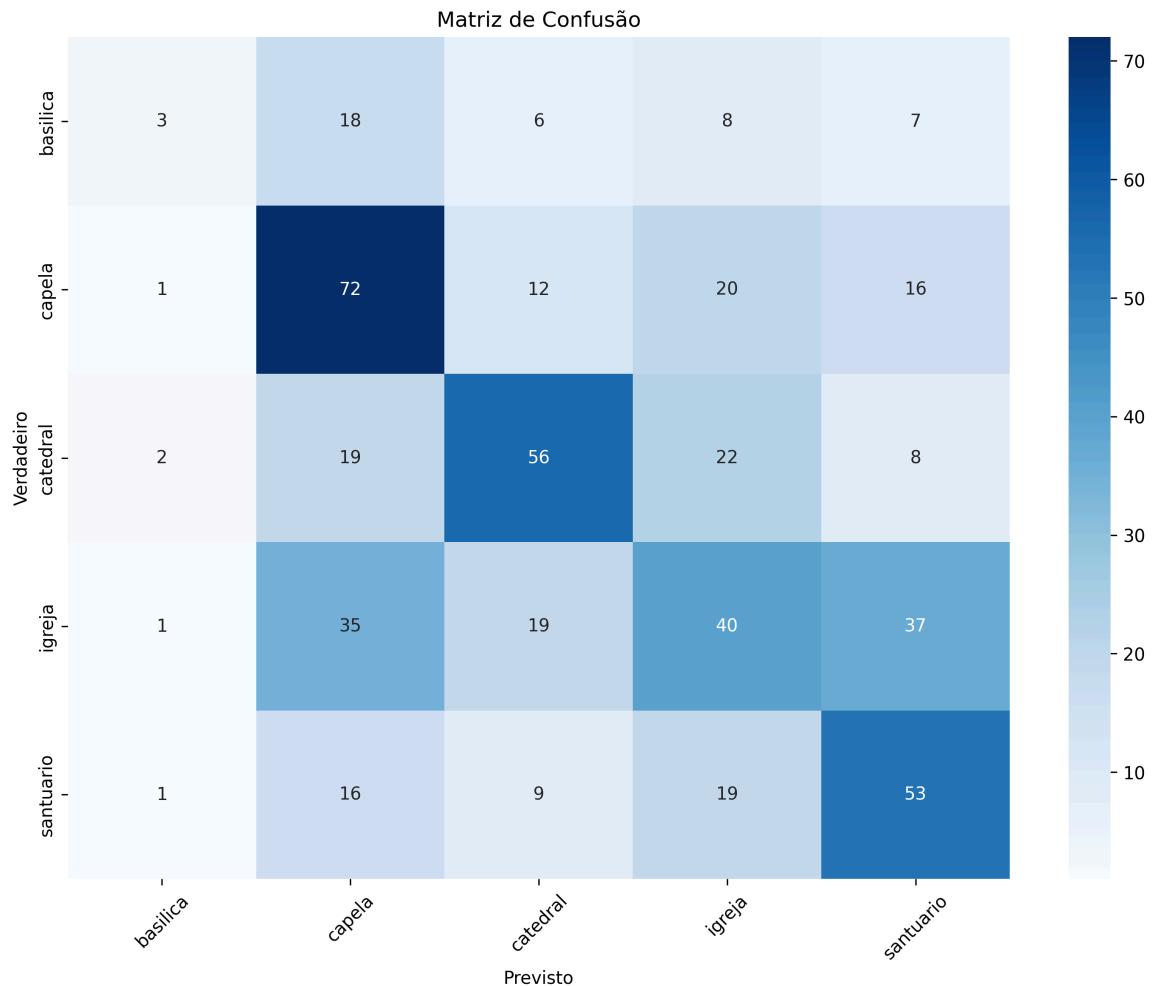


Figura 15: Matriz de confusão - Problema (2)

Anexo D - Tentativas falhadas com redes siamesas no decorrer do *pipeline*

Similaridade entre pares de imagens



Figura 16: Similaridade entre pares de imagens com valores muito próximos (TensorFlow)

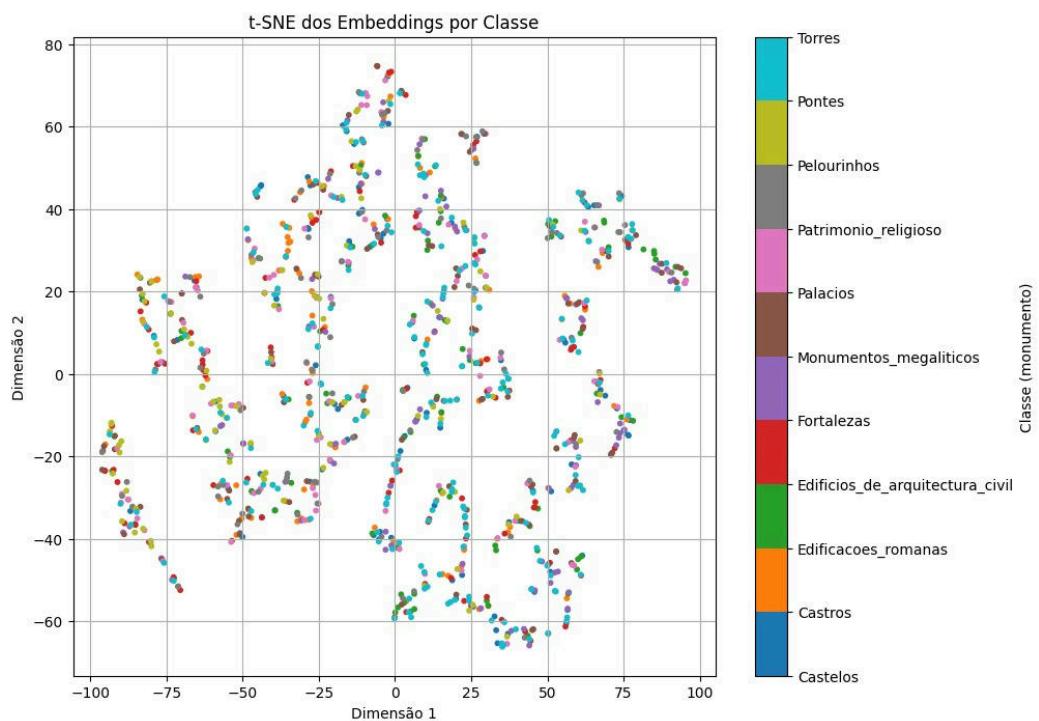


Figura 17: t-SNE dos *Embeddings*

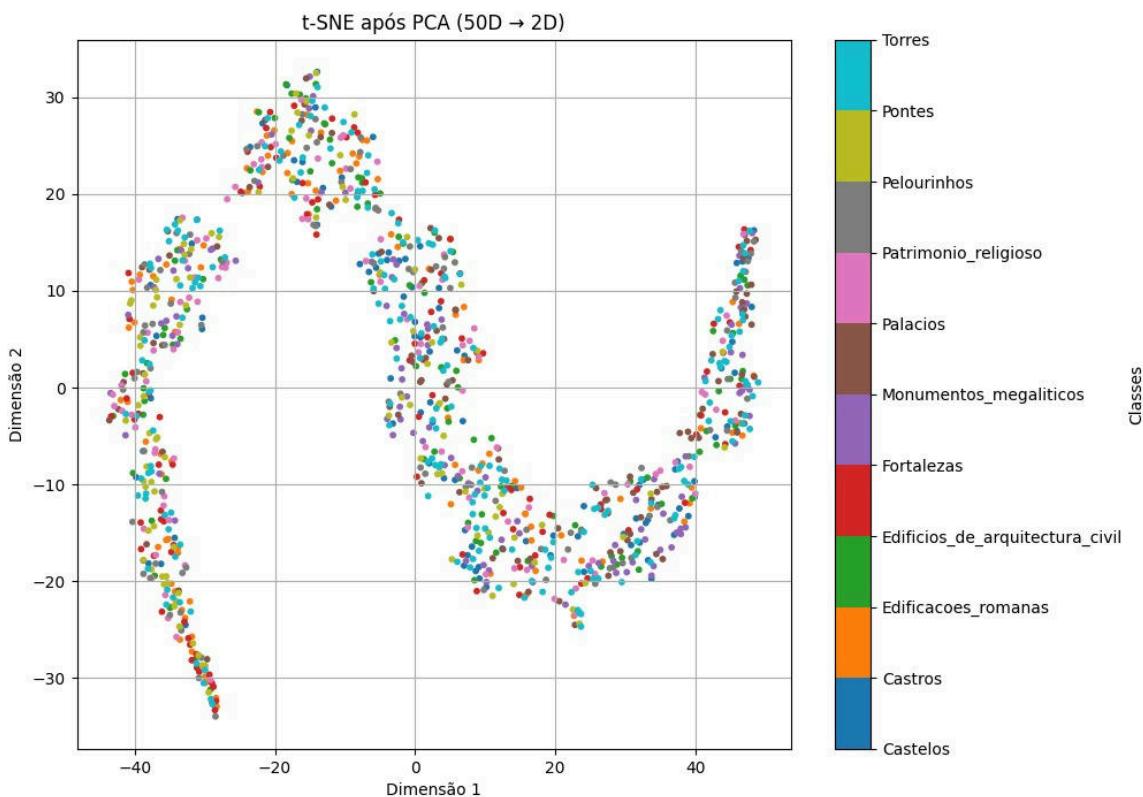


Figura 18: t-SNE dos *Embeddings*

Anexo E - Desenvolvimento do projeto - GitHub

“O repositório com todo o desenvolvimento do projeto (*scripts*, a imagem do Docker para o TensorFlow e as imagens resultantes dos *outputs*) pode ser consultado no GitHub através do seguinte [link](#).”

Streamlit como ideia para demo na apresentação do projeto

Work in progress...

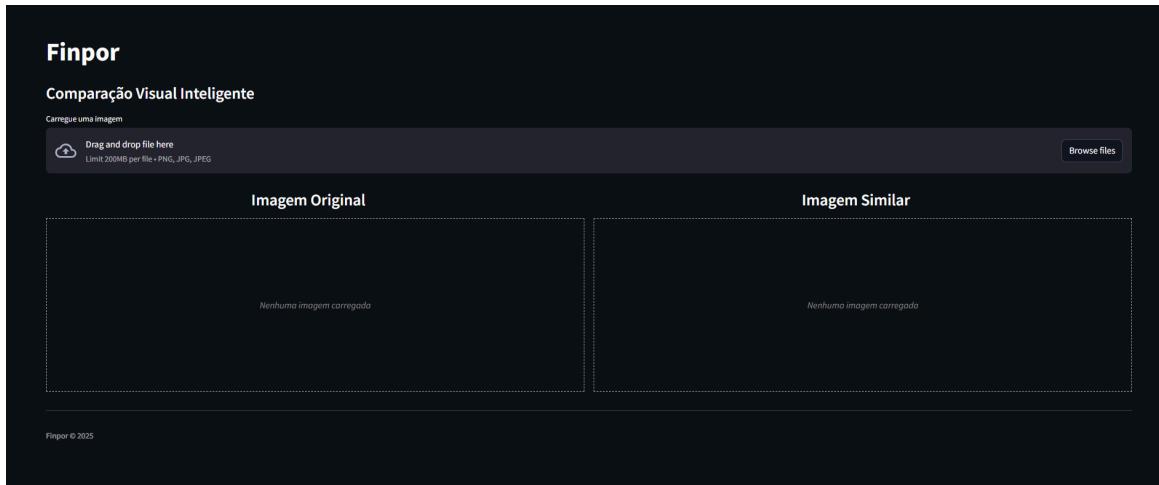


Figura 19: *Streamlit* para a apresentação do projeto