FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

Avaliação de Qualidade de Nuvens de Pontos baseada em Aprendizagem Profunda

Carlos Rafael Lopes Duarte

Coimbra, Julho 2019

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Đ
COIMBRA

Avaliação de Qualidade de Nuvens de Pontos baseada em Aprendizagem Profunda

Orientador:

Prof. Luís Alberto da Silva Cruz

Coimbra, Julho 2019

Agradecimentos

No fim deste percurso na Universidade de Coimbra falta-me agradecer a quem fez parte dele e me apoiou nestes últimos três anos.

Ao professor Doutor Luís Cruz, meu orientador de dissertação e dos vários projetos de investigação em que colaborei, por todas as ideias, conselhos, dicas, ensinamentos, mas sobretudo pela transmissão de conhecimento e disponibilidade para ensinar durante estes últimos dois anos.

A todos os participantes nos Testes de Avaliação de Qualidade Subjetiva de Nuvens de Pontos realizados no Laboratório de Processamento de Imagem do Departamento de Engenharia Electrotécnia e Computadores pelo contributo dado a esta dissertação e aos vários projetos de investigação relacionados.

A todos os meus professores que, de alguma forma contribuíram para a conclusão do meu percurso académico.

Aos meus colegas e amigos brasileiros que partilharam o Laboratório de Processamento de Imagem comigo, Mateus Grellert, Cristiano Santos e Iago Storch pelo conhecimento e ensinamentos transmitidos, pela amizade, pela paciência e pelo o apoio incondicional, um bem-haja a todos.

Ao meu amigo e colega, Tiago Oliveira o meu companheiro de percurso, pela amizade e companheirismo que sempre me deu ao longo destes três anos.

A todos os outros colegas que partilharam o Laboratório de Processamento de Imagem comigo, pelos conselhos e sugestões.

Aos meus pais, Carlos Duarte e Irene Duarte pelo amor de toda uma vida, pelo apoio incondicional, pelos conselhos certos na hora certa e toda a força que me deram para eu conseguir agarrar esta oportunidade de me formar. É graças a eles que sou o que sou.

À minha irmã Telma Duarte pelo amor, carinho, incentivo e força que só ela sabe dar. Peça fundamental na minha vida.

Às minhas sobrinhas que apareceram no meio deste percurso académico e que alegram a minha vida todos os dias. São a felicidade em estado puro.

À minha namorada Inês Oliveira que eu conheci na minha passagem por Coimbra, o meu porto de abrigo. Obrigado por todo o amor, apoio incondicional, sorrisos e abraços. Obrigado por seres quem és e por nunca me teres falhado.

A todos os meus amigos que Coimbra me apresentou pelo companheirismo e pela amizade que me deram, por todas as noites de festa e dias de estudo, foram eles que estiveram sempre presentes.

A todos os meus amigos de sempre e para sempre, esses sabem o que valem e sabem que são das pessoas

mais importantes da minha vida.

A todos estes o meu obrigado do fundo do coração.

Resumo

A tecnologia de Nuvens de Pontos tem sido uma das mais promissoras e mais exploradas no que toca à representação de objetos e mapeamento 3D. No entanto, a avaliação de qualidade visual deste tipo de conteúdos ainda não é considerada satisfatória.

Esta dissertação propõe um novo método de Avaliação Objetiva de Qualidade de Nuvens de Pontos baseada em resultados de Avaliação Subjetiva de Qualidade, utilizando Aprendizagem Profunda, particularmente Redes Neuronais Convolucionais.

Este trabalho consistiu em retreinar Redes Neuronais Convolucionais pré-treinadas, usando *Transfer Learning*, para prever de forma objetiva a qualidade deste tipo de conteúdos. Para isto foi necessário encontrar formas alternativas de representação de Nuvens de Pontos para que estas, em conjunto com resultados de Avaliação Subjetiva de Qualidade, pudessem ser utilizados para treinar as Redes Neuronais Convolucionais.

Por fim, após o treino das redes, foi criado um *dataset* suplementar de teste para confirmar a validade e qualidade dos resultados. As redes treinadas foram então utilizadas para prever os valores de qualidade para o *dataset* suplementar. Para os valores preditos pela rede, foram calculados fatores de desempenho que os comparam com os resultados subjetivos considerando estes últimos como referência.

Os melhores resultados obtidos para estes conteúdos foram: 0.2601 de Raíz do Erro Médio Quadrático (*Root Mean Square Error* - RMSE), 0.9822 de Coeficiente de Correlação de Pearson (*Pearson Correlation Coefficient* - PCC), 0.9137 de Coeficiente de Correlação de Spearman (*Spearman's Rank Order Correlation Coefficient* - SROCC) e 0.2778 de Rácio de *Outliers* (*Outlier Ratio* - OR).

Os resultados obtidos mostram que é possível utilizar Aprendizagem Profunda para avaliar a qualidade visual destes conteúdos, superando os resultados de estado da arte presentes na literatura.

Palavras-chave: Nuvens de Pontos, Avaliação Objetiva de Qualidade, Avaliação Subjetiva de Qualidade, Aprendizagem Profunda, *Transfer Learning*, Redes Neuronais Convolucionais

Abstract

Point Clouds has been one of the most promising and explored technologies regarding 3D object representation and 3D mapping. However, the solutions to automatically evaluate the visual quality of this type of contents are not yet satisfactory.

This Master Thesis propose a new methodology for Objective Quality Assessment of Point Clouds based on Subjective Quality Assessment results, using Deep Learning, particularly Convolutional Neural Networks.

The goal of this work was to retrain pre-trained Convolutional Neural Networks, using Transfer Learning, to predict the visual quality of this type of contents. For this it was necessary to find new alternative ways to represent Point Clouds to, together with Subjective Quality Assessment results, train the Convolutional Neural Networks.

Finally, after training the networks, an additional dataset was created to confirm the quality of the obtained results. The trained networks were used to predict the quality scores for the additional dataset. After this, some performance indexes were computed comparing the predicted scores against the subjective ground truth.

The best results to these contents were 0.2601 of Root Mean Square Error (RMSE), 0.9822 of Pearson Correlation Coefficient (PCC), 0.9137 of Spearman's Rank Order Correlation Coefficient and 0.2778 of Outlier Ratio (OR).

The results obtained show that is possible to predict the subjective visual quality of Point Clouds using Deep Learning, outperforming state-of-art results.

Keywords: Point Clouds, Objective Quality Assessment, Subjective Quality Assessment, Deep Learning, Transfer Learning, Convolutional Neural Networks

“ It is strange that only extraordinary men make the discoveries, which later appear so easy and simple."

— Georg C. Lichtenberg

Conteúdo

Lista de Acrónimos e Siglas

3DTK	*3D Toolkit*
CNN	Rede Neuronal Convolucional *(Convolucional Neural Network)*
DSIS	*Double Stimulus Impairment Scale*
FPS	*Frames Per Second*
FR	*Full-Reference*
GPU	*Graphics Processing Unit*
H.264/AVC	*H.264 / Advanced Video Coding*
ILSVRC	*ImageNet Large Scale Visual Recognition Competition*
IP	*Internet Protocol*
ITU	*International Telecommunications Union*
JPEG	*Joint Photographic Experts Group*
ML	*Machine Learning*
MOS	*Mean Opinion Scores*
MPEG	*Moving Picture Experts Group*
MSSIM	*Mean Structural Similarity Measure*
NR	*No-Reference*
OR	*Outlier Ratio*
PC	Nuvem de Pontos (*Point Cloud*)
PCC	*Pearson Correlation Coeficient*
PCL	*Point Cloud Library*
PSNR	*Peak Signal-to-Noise Ratio*
PSNR-HVS	*Peak Signal-to-Noise Ratio - Human Visual System*
PSNR-HVS-M	*Peak Signal-to-Noise Ratio - Human Visual System - Visual Masking*

ReLU	Unidade de Retificação Linear *(Rectified Linear Unit)*
RGB	*Red Green Blue*
RMSD	*Root Mean Square Distance*
RMSE	*Root Mean Square Error*
RR	*Reduced-Reference*
SROCC	*Spearman's Rank Correlation Coefficient*
SSIM	*Structural Similarity*
VIFP	*Visual Information Fidelity Pixel-based*

Lista de Figuras

Lista de Tabelas

1 Introdução

Vivemos na era do vídeo e da imagem, como o prova [1] que apresenta previsões segundo as quais em 2022 a percentagem de todo o tráfego IP dedicado ao vídeo será de 82%. O avanço tecnológico atual dá passos de gigante o que faz com que os utilizadores sejam mais exigentes obrigando a tecnologia a uma constante evolução, isto em todas as áreas e o vídeo não foge a este paradigma. Isto leva-nos à necessidade crescente de vídeo e imagem cada vez mais realistas e imersivas que representem todos os detalhes com excelente qualidade.

Muitas tecnologias de representação 3D foram testadas, contudo nem todas alcançaram grande sucesso. Uma das mais promissoras é a tecnologia de Nuvens de Pontos (*Point Clouds* - PCs), já com muitas aplicações práticas desde os videojogos até à condução autónoma de veículos. Esta representação tem-se revelado útil e fiável na representação de objetos e mapeamento 3D. Como todas as outras representações também as PCs têm desvantagens, particularmente o volume de dados que ocupam. Estas nuvens podem conter milhões de pontos, milhões de pontos que ocupam milhões de bits, pelo que a transmissão destes conteúdos sem qualquer tipo de compressão é uma tarefa impossível.

Deste modo, torna-se necessário implementar técnicas de compressão à semelhança do que é feito em vídeo 2D. Aplicar compressão quase sempre significa degradação da qualidade visual, e em PCs pode significar, amostragem de pontos, adição de resíduos, deslocação de pontos, adição de distorção, entre outros. É então necessário obter uma forma de avaliar a qualidade visual das PCs comprimidas de uma forma automática para garantir que o utilizador tenha a melhor experiência possível.

Existem algumas métricas de avaliação de qualidade para este tipo de informação mas ainda não estimam de forma fiável a qualidade das PCs, um problema que para vídeo 2D tem algumas soluções, não sendo o caso em nuvens de pontos. O propósito desta dissertação é encontrar melhores formas de estimação de qualidade de maneira automática e fiável.

1.1 Objetivos desta Dissertação

O objetivo deste trabalho é criar uma nova forma de avaliar a qualidade visual objetiva de Nuvens de pontos que se correlacione bem com resultados de avaliações de qualidade visual subjetiva que são considerados de referência. Para isto irá ser utilizada aprendizagem profunda, particularmente Redes Neuronais Convolucionais (*Convolucional Neural Networks* - CNNs).

Estas redes já provaram a sua utilidade em análise de imagem 2D, particularmente em problemas de classificação [2]. Para além disto também existem estudos de avaliação de qualidade visual objetiva de imagens 2D com bons resultados que utilizam este tipo de redes, como demonstram os autores de [3] e de

[4].

A primeira parte deste trabalho passa por criar uma forma eficaz de representar nuvens de pontos que seja compatível com a entrada de CNNs já estudadas. Obtidos estes dados de entrada *(dataset)* irão ser testadas e retreinadas várias CNNs usadas para classificação de imagens, usando *transfer learning*. Estas redes irão sofrer ligeiras alterações nas suas arquiteturas e nos seus hiper-parâmetros. Isto com o intuito de adaptar estas redes a um problema de regressão, tal como é avaliação de objetiva de qualidade. Associado a este *dataset* irão ser utilizados resultados provenientes de estudos de avaliação subjetiva de qualidade visual de PCs para que as CNNs sejam treinadas tendo como referência estes resultados.

1.2 Organização do texto

No capítulo 2 estão sumarizados conceitos básicos sobre nuvens de pontos e sobre a sua representação.

O capítulo 3 descreve as várias formas de avaliação de qualidade disponíveis para conteúdos digitais particularmente para nuvens de pontos. Também apresenta estudos de avaliação de qualidade subjetiva de PCs que foram realizados na Universidade de Coimbra em conjunto com outras universidades.

O capítulo 4 contém os conceitos básicos sobre redes neuronais convolucionais, exemplos de arquiteturas de várias CNNs e CNNs que foram utilizadas para medida de objetiva de qualidade.

O capítulo 5 descreve todo o trabalho que foi feito, particularmente a construção do *dataset*, a atribuição de etiquetas, definição de hiper-parâmetros e modificações na arquitetura das redes.

O capítulo 6 descreve o treino das redes e todos os resultados obtidos experimentalmente para as várias redes e modificações testadas.

O capítulo 7 conclui este trabalho sumarizando todos os resultados obtidos e possíveis melhorias que possam vir a ser implementadas.

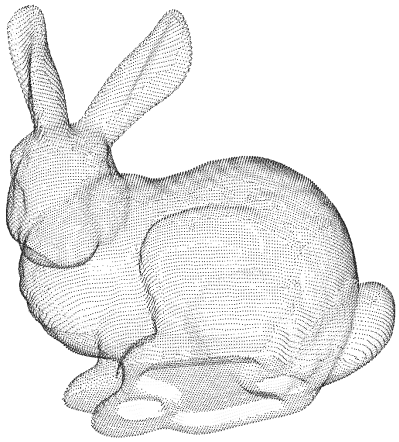
2 Nuvens de Pontos

2.1 Conceitos básicos sobre Nuvens de Pontos

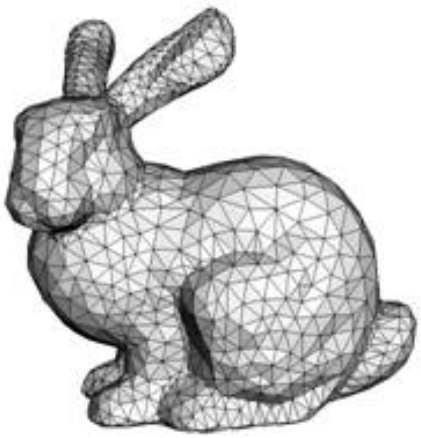
Uma PC é um conjunto de pontos com informação de posição associada expressa num determinado sistema de coordenadas, tipicamente um sistema 3D em que a posição de cada ponto é representada por um conjunto de três valores reais (x,y,z). Frequentemente estas PCs representam a superfície externa de um objeto ou o mapeamento de algum local. Esta informação de posição pode ser complementada com informação de textura (RGB) [5], informação dos vetores normais e, para alguns casos especiais informações de temperatura, entre outros.

Os pontos das PCs tanto podem ser observados diretamente num sistema de coordenadas 3D ou podem sofrer um pré-processamento de forma a tornar mais percetível o seu formato. É o exemplo das *Meshes*, que são uma conversão dos pontos da PC numa malha de polígonos interligados, como ilustra a figura 2.1b.

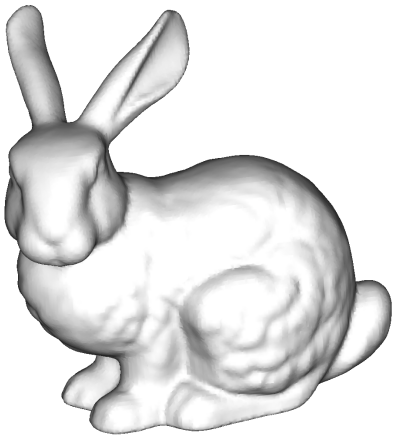
Na prática *Meshes* 3D ligam os pontos da PC que irão tornar-se nos vértices das faces da *Mesh*. A textura destas faces pode ser obtida interpolando a informação de cor dos vértices das mesmas. Geralmente *Meshes* são mais fáceis de percecionar que os pontos isolados de uma PC mas a complexidade da conversão para *Meshes* é um problema a ter em conta para aplicações em tempo real [5]. Para além das *Meshes* existem outros métodos de representação que facilitam a visualização de PCs, particularmente algoritmos de reconstrução de superfícies, que a partir dos pontos das PCs constroem uma superfície que melhor se ajusta à PC. O algoritmo mais utilizado é o algoritmo de reconstrução de *Poisson* [6] pelo facto de, geralmente, as superfícies obtidas com este algoritmo representarem de forma clara os objetos presentes nas PCs, ilustrado na figura 2.1c.



(a) Nuvem de pontos



(b) *Mesh*



(c) Nuvem de pontos reconstruída com o algoritmo de *Poisson*

Figura 2.1: Nuvem de pontos "Bunny"de *Standford 3D Scanning repository.*

2.2 Vetores normais

Nuvens de pontos não são mais que um conjunto de pontos com coordenadas associadas, pelo que, é possível aplicar cálculos geométricos a este tipo de conteúdos. Para facilitar o processamento deste tipo de dados foi introduzido o conceito de vetores normais a PCs. Um vetor normal é um vetor que é ortogonal a um plano. Um plano pode ser definido por três pontos não colineares, pelo que, para uma PC podemos definir localmente, para um conjunto restrito de pontos vizinhos, o plano que melhor se ajusta a eles. Para calcular os vetores normais em cada ponto de uma PC é necessário definir o número de pontos vizinhos que se vão ter em conta, calcular o plano que melhor se ajusta a eles e o seu respetivo vetor normal. Este cálculo encontra-se descrito em [7]. Com a obtenção de vetores normais é possível caracterizar a PC de uma forma mais completa. Estes vetores são usados em inúmeras aplicações como por exemplo algoritmos de reconstrução de superfícies de PCs e em métricas de avaliação de qualidade visual objetiva de PCs. A figura 2.2 ilustra a PC *bunny* com os seus vetores normais.

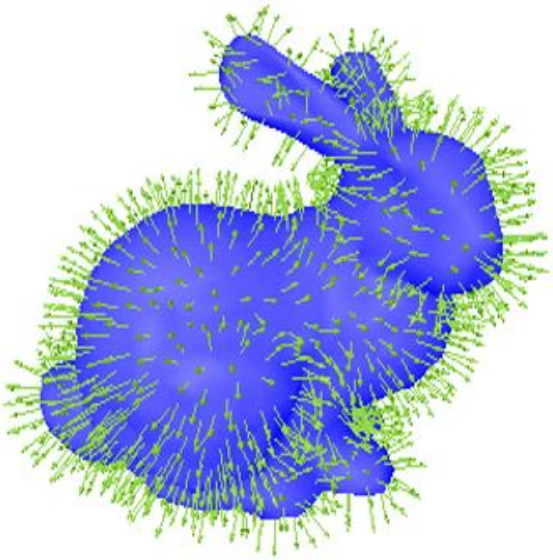


Figura 2.2: PC *bunny* com os respetivos vetores normais (Fonte: [8]).

3 Avaliação de Qualidade

3.1 Avaliação Objetiva de Qualidade

O propósito da investigação em avaliação objetiva de qualidade é criar modelos computacionais que permitam estimar a qualidade que o utilizador perceciona, ou seja a qualidade subjetiva. Estas medidas numéricas só são úteis se apresentarem uma boa correlação com a impressão subjetiva de qualidade de um observador humano. Por outras palavras o modelo deverá ser capaz de prever a qualidade que um ser humano normal perceciona ao visualizar o conteúdo [9].

Um dos critérios para classificar medidas de avaliação objetiva de qualidade é o uso do conteúdo original como referência. Se o modelo computacional se basear na comparação do conteúdo original com o conteúdo degradado será uma métrica do tipo *full-reference* (FR), ilustrado na figura 3.1. Se pelo contrário não houver acesso ao conteúdo original usado como referência a medida é classificada como *no-reference* (NR), representado na figura 3.2. A medida pode ainda ser classificada como *reduced-reference* (RR) se apenas estiverem disponíveis algumas caraterísticas do conteúdo original e não o conteúdo original no seu todo, tal como ilustra a figura 3.3.

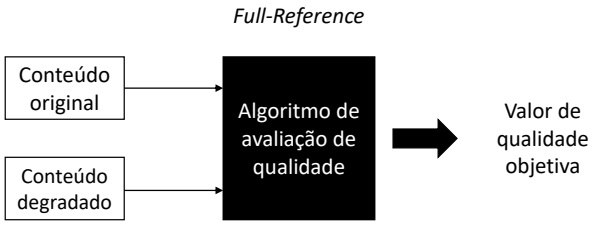


Figura 3.1: Métrica do tipo *full-reference*.

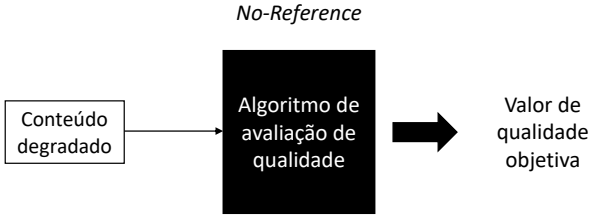


Figura 3.2: Métrica do tipo *no-reference*.

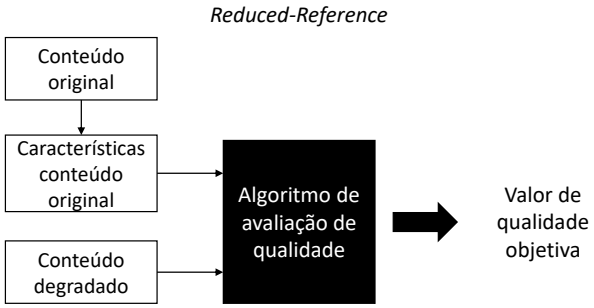


Figura 3.3: Métrica do tipo *reduced-reference*.

3.2 Avaliação Objetiva de Qualidade de Nuvens de Pontos

Como sucede com outros conteúdos digitais, em geral o uso de compressão implica perda de qualidade visual. Em determinadas circunstâncias é importante saber avaliar o impacto do grau de compressão na qualidade de uma PC após a compressão.

Avaliar a qualidade de uma PC processada ou degradada é um problema bastante importante para o qual ainda não foi encontrada uma solução satisfatória.

A qualidade pode ter em conta atributos dos pontos da PC para além da sua informação geométrica (i.e posição no espaço) mas tem sido dado mais ênfase a esta última no que toca à avaliação de qualidade de PCs [10].

Todas as medidas presentes na literatura que avaliam a qualidade objetiva de PCs seguem a metodologia FR. As duas métricas principais usadas para medir distorção geométrica de PCs são a distância ponto-a-ponto e a distância ponto-a-plano. Para além destas duas, existem outras métricas mais recentes que provaram conseguir melhores resultados do que as referidas anteriormente. Uma delas é baseada em similaridade angular entre os vetores normais da PC degradada e da PC original e outra baseada nas projeções em 2D das PCs. Existem também outros métodos para medir distâncias entre nuvens de pontos nomeadamente os algoritmos ponto-a-superfície e superfície-a-superfície.

3.2.1 Métrica ponto-a-ponto

Esta métrica tem por base a média das distâncias entre pontos da PC degradada e os seus correspondentes na nuvem original. Para cada ponto na PC degradada é encontrado o seu ponto vizinho mais próximo na nuvem original e calculada a distância entre ambos. Na figura 3.4 mostra-se como é obtida esta distância. Os círculos correspondem à PC "A"(degradada) e as circunferências correspondem à PC "B"(de referência). O círculo a vermelho é o ponto na PC "A"cujo correspondente é a circunferência a vermelho na PC "B", sendo $d(A_i, B_i)$ a distância entre eles.

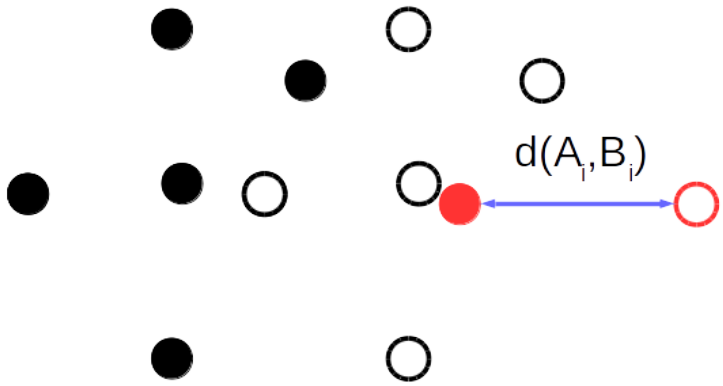


Figura 3.4: Distância ponto-a-ponto ($d(A_i, B_i)$).

Estas distâncias ponto-a-ponto são usadas para calcular a distorção global de duas maneiras diferentes [11]:

- Calculando a raiz da média quadrática das distâncias, *Root Mean Square Distance* (RMSD) ou *Root Mean Square Error* (RMSE), adicionando a magnitude dos vetores de distância e calculando a raiz quadrada do resultado dividido pelo número de pontos da PC degradada (expressão 3.1). É também comum o uso do *Mean Square Distance* (MSD) ou *Mean Square Error* (MSE) que é em tudo igual ao RMSE mas sem o cálculo da raiz quadrada (expressão 3.2).

$$d_{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} d(A_i, B_i)^2}{N}} \tag{3.1}$$

$$d_{MSE} = \frac{\sum_{i=1}^{N} d(A_i, B_i)^2}{N} \tag{3.2}$$

- Calculando a distância de Hausdorff (i.e máximo de todas as distâncias) do conjunto de pontos na PC degradada e os seus correspondentes na PC de referência.

$$d_{Hausdorff} = \max_{\forall i \in A}(d(A_i, B_i)) \tag{3.3}$$

3.2.2 Métrica ponto-a-plano

Esta métrica é ligeiramente mais complexa que a distância ponto-a-ponto pois envolve o cálculo de vetores normais para conseguir aferir a distância ponto-a-plano, e segue os seguintes passos:

- Para cada ponto na PC "A"original é identificado o seu correspondente na PC "B"degradada.
- Para esse mesmo ponto na PC "A"o seu vetor normal unitário é usado se disponível, se não, o vetor normal é estimado usando os pontos vizinhos mais próximos do ponto a ser avaliado [7].
- Depois disto o vetor da distância ponto-a-ponto *E(i,j)* é calculado entre os pontos das diferentes PCs e projetado sobre a direção do vetor normal tal como é demonstrado na figura 3.5.

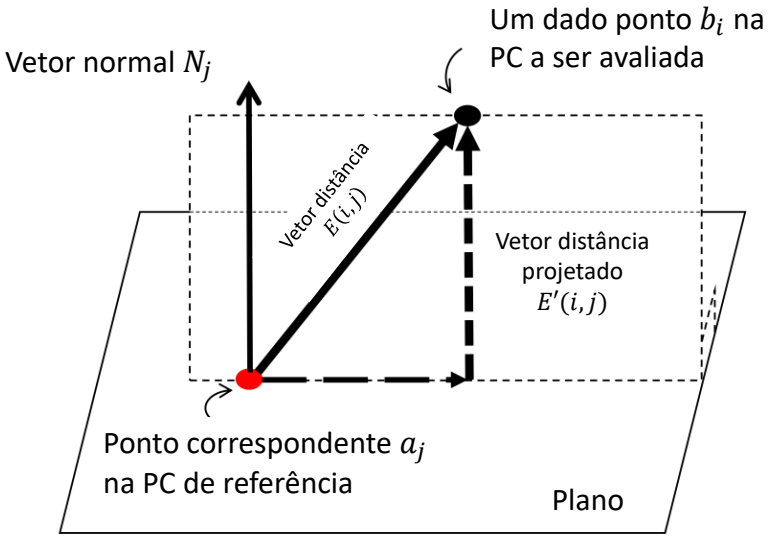


Figura 3.5: Distância ponto-a-ponto vs ponto-a-plano (Adaptada de [12]).

- A medida final é obtida através do MSE de todas as magnitudes dos vetores projetados (expressão 3.4) [12].

$$d_{p2plane(MSE)} = \frac{\sum_{i=1}^{N}(E(i,j) \cdot N_j)^2}{N} \tag{3.4}$$

Em comparação com a métrica ponto-a-ponto a métrica ponto-a-plano apresenta uma melhor correlação com resultados subjetivos.

3.2.3 Métrica baseada em similaridade angular

Esta é uma das métricas propostas recentemente e que apresenta melhor correlação com experiências subjetivas [13]. Esta tem por base a tendência natural que o cérebro humano tem em interpolar pontos de maneira a compreender a geometria de um objeto. Usando este efeito esta medida pode ser obtida estimando as semelhanças entre superfícies ajustadas às PCs e segue os seguintes passos:

- Para cada ponto da PC de referência é encontrado o seu correspondente na nuvem a analisar. De seguida para cada um desses pontos é calculado um plano tangente à superfície da PC e respetivo vetor normal. Com esta informação é possível calcular o ângulo θ entre essas normais ilustrado na figura 3.6 e, com base nesse ângulo calcular o grau de similaridade das duas superfícies que passam por esses dois pontos, sendo que, a semelhança máxima é quando $\theta = 0°$.

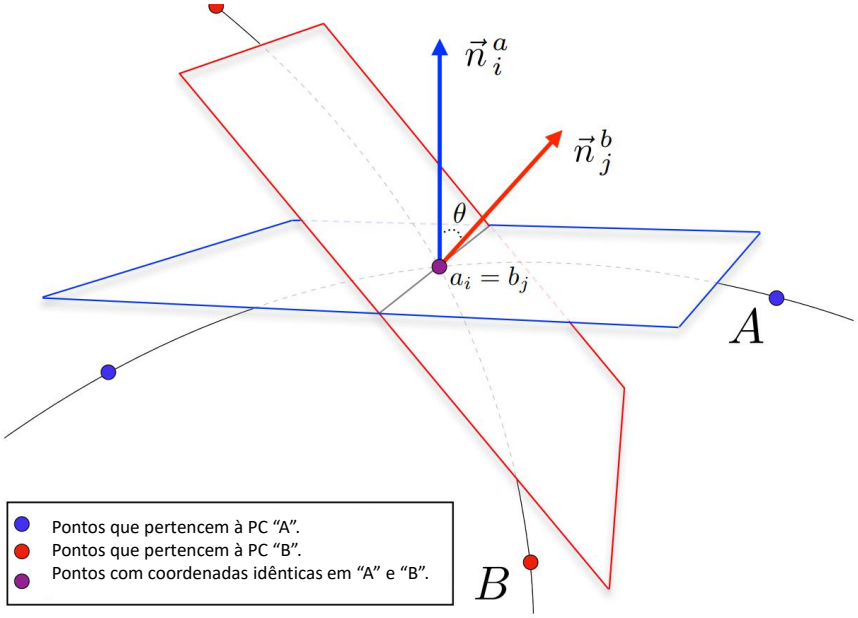


Figura 3.6: Métrica de similaridade angular (Adaptada de [13]).

- Após calculado este grau de similaridade para cada ponto da PC de referência, é calculada a média de todos esses valores. Isto é feito primeiro da nuvem de referência para a degradada e também vice-versa. Depois destes dois valores de média obtidos o valor da métrica é obtido escolhendo o valor mínimo desses dois valores.

3.2.4 Métrica com base em projeções ortogonais 2D

Este tipo de avaliação de qualidade proposta recentemente e com boa correlação com resultados subjetivos [14], baseia-se nas projeções de uma nuvem de pontos sobre as várias faces de um cubo no qual se inscreve toda a PC. Uma vez obtidas essas projeções são usadas várias métricas de avaliação qualidade objetiva para imagem 2D aplicadas a cada par de projeções, da PC original e da PC degradada. Entre outras têm sido usadas como medidas de qualidade 2D, PSNR, PSNR-HVS, PSNR-HVS-M, SSIM, MSSIM e VIFP. O algoritmo, usando como exemplo o PSNR, está ilustrado na figura 3.7. Segundo [14] os melhores resultados são obtidos com MSSIM e VIFP.

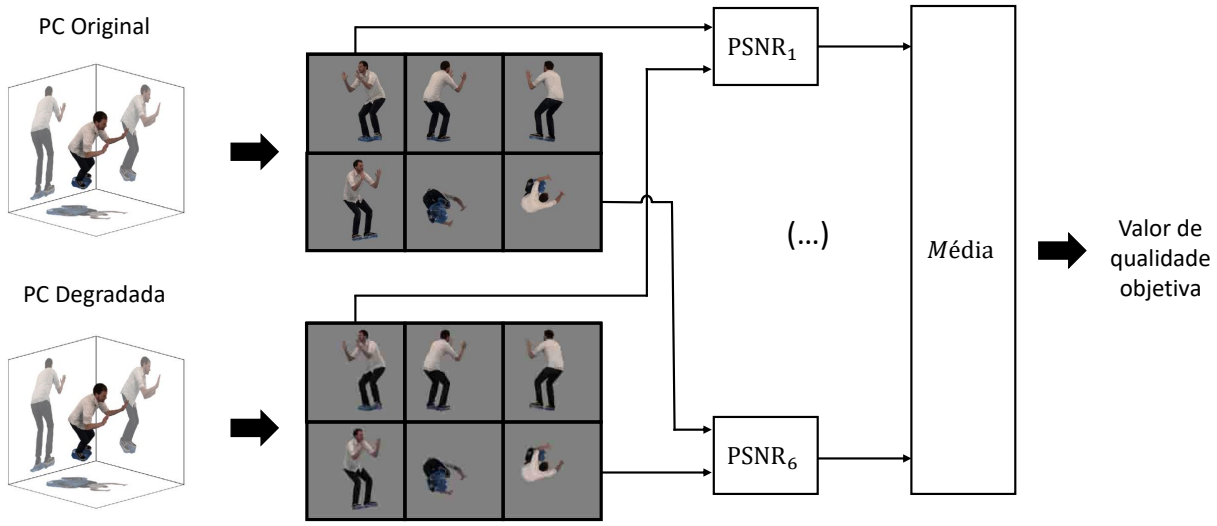


Figura 3.7: Diagrama do algoritmo da métrica com base em projeções ortogonais 2D (Adaptada de [15]).

3.3 Avaliação Subjetiva de Qualidade

Como referido anteriormente, avaliação de qualidade objetiva só pode ser considerada útil se representar de uma forma fiável a perceção visual humana. Para isto é necessário introduzir o conceito de avaliação de qualidade subjetiva que tem como objetivo recolher dados reais de perceção visual humana.

As avaliações subjetivas dividem-se em dois tipos, avaliações de qualidade (*quality assessments*) e avaliações de degradação (*impairment assessments*). As avaliações de qualidade avaliam de uma forma absoluta a qualidade de um conteúdo (sem qualquer tipo de comparação com outro conteúdo). Este tipo de avaliações é normalmente utilizado quando não existe um conteúdo de referência disponível. Os conteúdos são avaliados usando a seguinte escala:

Tabela 3.1: Escala avaliações de qualidade (Fonte: [16]).

5	Excelente *(Excellent)*
4	Bom *(Good)*
3	Satisfatório *(Fair)*
2	Fraco *(Poor)*
1	Mau *(Bad)*

As avaliações de degradação avaliam a qualidade de um conteúdo relativamente a outro, tipicamente usando uma escala de degradação que avalia as diferenças entre os conteúdos como a seguinte:

Tabela 3.2: Escala avaliações de degradação (Fonte: [16]).

5	Impercetível *(Imperceptible)*
4	Percetível, mas não irritante *(Perceptible, but not annoying)*
3	Ligeiramente irritante *(Slightly annoying)*
2	Irritante *(Annoying)*
1	Muito irritante *(Very annoying)*

Para realizar testes subjetivos apropriados é necessário selecionar a partir das opções disponíveis aqueles que mais se adequam ao contexto, ao objetivo e às circunstâncias em que os conteúdos a ser testados vão ser observados.

Os observadores podem ser classificados em *especialistas* e *não especialistas*, sendo que um observador *não especialista* é um observador que não tem qualquer conhecimento sobre o conteúdo do sistema em teste. Em qualquer dos casos os observadores nunca deverão estar envolvidos na preparação da experiência que está a ser realizada.

A acuidade visual e a normal perceção das cores destes observadores devem ser previamente verificadas e devem ser utilizados pelo menos 15 observadores [16].

A sessão deve ter uma duração máxima de meia hora. No início da experiência devem ser introduzidas cerca de cinco apresentações de treino (*dummy presentations*) de maneira a estabilizar a opinião dos observadores. Os dados provenientes destas apresentações não devem ser tidos em consideração na análise final de resultados. As apresentações devem ter uma ordem aleatória diferente para cada observador [16].

Para avaliação de qualidade de PCs o método mais utilizado é o *double-stimulus impairment scale (DSIS) simultaneous* com uma escala de de 5 níveis (a da tabela 3.2) para avaliar o grau de diferença entre os conteúdos. Neste método é pedido ao observador para ver um par de imagens ou vídeos lado a lado em simultâneo, em que uma é o conteúdo original e outra é o mesmo conteúdo mas com um nível de degradação associado. Depois de ver as imagens ou vídeos é pedido ao observador que avalie as diferenças entre o conteúdo original e o conteúdo degradado utilizando a escala da tabela 3.2.

O escalonamento temporal deste método está ilustrado na figura 3.8. Em que T1 corresponde ao intervalo de tempo da primeira apresentação do par de imagens ou vídeos, T2 corresponde ao intervalo de tempo de fundo neutro (cinza ou preto) e T3 corresponde ao intervalo de tempo da segunda apresentação do par de imagens ou vídeos. Cada observador tem de avaliar os conteúdos durante o intervalo de tempo em que decorre a apresentação dos mesmos e o instante de tempo de fundo neutro imediatamente a seguir. Por exemplo, para avaliar os conteúdos presentes em T1 o observador tem o intervalo de tempo de T1+T2 para o fazer. A experiência não avança enquanto o observador não avaliar o conteúdo visualizado.

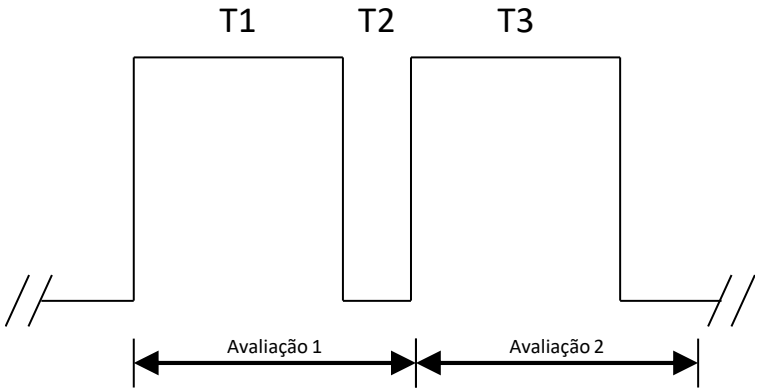


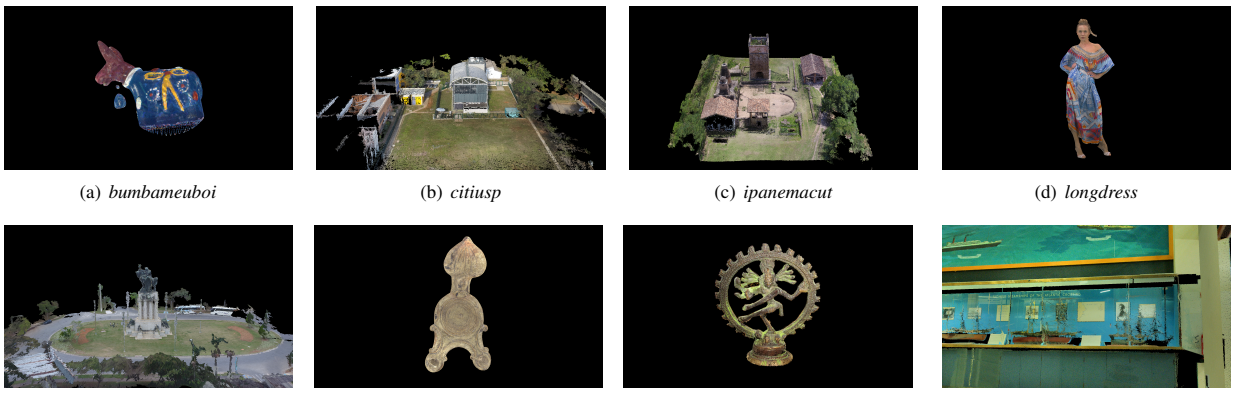
Figura 3.8: Exemplo de escalonamento temporal para o método *DSIS simultaneous*.

3.4 Teste de Avaliação Subjetiva de Qualidade de Nuvens de Pontos com Textura

No âmbito do projeto JPEG-PLENO foram realizadas várias sessões de avaliação de qualidade Subjetiva com PCs com informação de cor associada. Estas experiências tiveram como intuito testar algoritmos de codificação de PCs com vários níveis de degradação. Este estudo foi realizado no âmbito de uma colaboração de várias universidades europeias [17].

3.4.1 Preparação dos conteúdos

Neste estudo foram utilizadas oito PCs com informação de geometria e textura associadas. Este estudo usou quatro PCs de pequena dimensão nomeadamente *bumbameuboi*, *longdress*, *romanoillamp* e *shiva* e quatro PCs de grande escala *ucl*, *citiusp*, *ipanemacut* e *ramos*, ilustradas na figura 3.9. Todos estes conteúdos fazem parte do *dataset* do projeto JPEG-PLENO [18].



(a) *bumbameuboi* (b) *citiusp* (c) *ipanemacut* (d) *longdress*

(e) *ramos* (f) *romanoillamp* (g) *shiva* (h) *ucl*

Figura 3.9: Vista frontal de cada uma das PCs de referência (Fonte: [17]).

Um dos métodos utilizados para comprimir estas PCs foi *octree pruning*, que consiste numa remoção de pontos regular distanciados por um valor máximo de erro conhecido. Este esquema de degradação foi implementado usando a *Point Cloud Library* [7]. O outro método utilizado é um codificador baseado em projeções implementado no *3DTV toolkit* [19].

Para cada um dos métodos as PCs foram comprimidas para três níveis de qualidade diferentes, um de qualidade baixa , uma de qualidade média e outro de qualidade elevada. Para o método de *octree pruning* o nível de qualidade foi definido com base na percentagem de pontos restantes da PC. No método que utiliza o *3DTK toolkit* os níveis de qualidade foram definidos de forma a que a percentagem de pontos restantes fosse similar ao obtido para o método de *octree pruning*. O número de pontos para cada conteúdo de referência e degradado, bem como outras informações como o tamanho do ponto usado na visualização das nuvens usando o *CloudCompare* [20] podem ser encontrados na tabela 3.3.

Tabela 3.3: Configurações de compressão e renderização (Fonte: [17]).

Conteúdo	**Tipo de compressão**	**Qualidade da compressão**	**# de pontos**	**% de pontos**	**Tamanho do ponto**
Bumbameuboi	**original**	–	150379	100	4
	3dtk	**Elevada**	68971	45.86	6
		Média	27796	18.48	7
		Baixa	8151	5.42	13
	pcl	**Elevada**	69197	46.02	6
		Média	27891	18.55	7
		Baixa	8229	5.47	13
Longdress	**original**	–	857966	100	2
	3dtk	**Elevada**	844980	98.49	2
		Média	254380	29.65	3
		Baixa	67045	7.81	4
	pcl	**Elevada**	857966	100	2
		Média	254322	29.64	3
		Baixa	66520	7.75	4
Romanoillamp	**original**	–	1286052	100	2
	3dtk	**Elevada**	635938	49.45	2
		Média	270663	21.05	3
		Baixa	76141	5.92	5
	pcl	**Elevada**	636008	49.45	2
		Média	270088	21	3
		Baixa	77104	6	5

Tanto para os conteúdos de pequena escala como para os de grande escala foram escolhidos tamanhos de ponto diferentes para cada tipo de conteúdo e de degradação. Para além disso também foram escolhidas diferentes distâncias da câmara à origem do referencial para os diferentes conteúdos. Foram testadas e selecionadas várias combinações destes parâmetros após uma verificação visual efetuada por vários peritos, por forma a disponibilizar ao utilizador a melhor perceção possível sobre o conteúdo.

Para uma completa visualização da PC original e da PC degradada em simultâneo sobre vários pontos de vista, foi necessário gerar vídeos que foram visualizados e avaliados com base na sua qualidade visual pelos participantes. Estes vídeos foram gerados com base em dois percursos de câmara distintos. Um para os conteúdos de pequena dimensão, que consistiu na rotação da câmara à volta do eixo vertical e de seguida à volta do eixo horizontal com passos de 1° (figura 3.10), e outro para os conteúdos de grande escala (com exceção da *ucl*), que consistiu apenas na rotação à volta do eixo vertical com passos de 0.5° (figura 3.11). Para a PC *ucl* foi necessário definir um percurso de navegação pelo cenário, que resultou num vídeo mais imersivo, pois para este conteúdo não foi possível definir um eixo de rotação da câmara. Para todos os casos a câmara foi colocada e movimentada de maneira a que os participantes conseguissem percecionar as PCs na sua totalidade. Com estes percursos da câmara foram obtidos 720 *frames* através do software CloudCompare

[20].

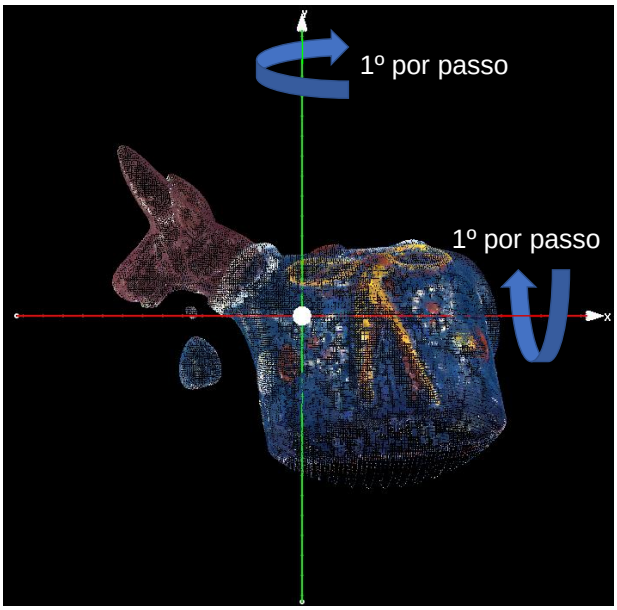


Figura 3.10: Esquema de rotação para PCs de pequena escala.

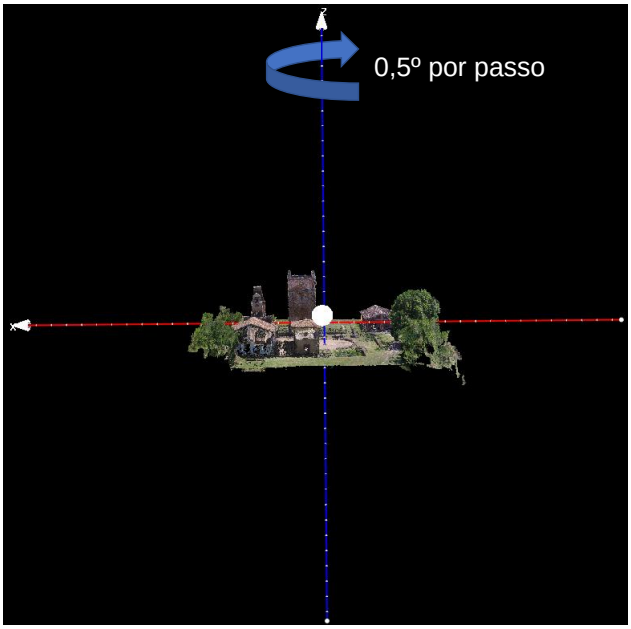


Figura 3.11: Esquema de rotação para PCs de grande escala.

Foram gerados um total de 720 *frames* de resolução de 1920x1080 para cada conteúdo. Estes *frames* foram então comprimidos sem perdas com um codificador H.264/AVC (usando o FFmpeg), produzindo um vídeo animado a 30 fps com uma duração total de 24 segundos. A cor de fundo escolhida foi o preto.

3.4.2 Metodologia de avaliação

As experiências subjetivas foram conduzidas em três laboratórios diferentes: Universidade da Beira Interior (UBI), Covilhã, Portugal; Universidade de Coimbra, Coimbra, Portugal e University North (UNIN), Varaždin, Croácia. As condições em que foram realizados estes testes seguem a recomendação BT.500-13

do ITU-R [16], em que foi adotado o método *DSIS simultaneous* com uma escala de avaliação de 5 níveis (tabela 3.2), incluindo uma referência escondida para controlo.

Os vídeos da PC de referência e da PC degradada foram colocados lado a lado e cada observador teria que avaliar consoante o grau de similaridade com o conteúdo de referência. Foram obtidos um total de 42 resultados por avaliação em que cada observador avaliou sete conteúdos diferentes com dois esquemas de compressão diferentes com três níveis distintos de qualidade, mais referências escondidas.

Os resultados obtidos foram filtrados para remover *outliers* tendo sido usado o método descrito na recomendação BT.500-13 do ITU-R [16], sendo que não foram encontrados nenhuns *outliers*. De seguida, foram obtidos os *mean opinion scores (MOS)* e os intervalos de confiança de 95% assumindo uma distribuição *T-Student's*. Este estudo contou com a participação total de 50 observadores.

3.4.3 Resultados

Os resultados subjetivos obtidos nesta experiência serviram essencialmente para aferir que tipo de esquema de compressão os participantes preferem, se o baseado em *octree* (PCL), ou o baseado em projeções (3DTK). Foi possível concluir que na generalidade os participantes preferem degradação baseada em *octree* ao invés da baseada em projeções.

Estes resultados subjetivos também serviram para estabelecer correlações com as métricas de avaliação objetiva de nuvens de pontos. Baseado na recomendação ITU-T P.1401 [21], o coeficiente de correlação de Pearson (PCC), o coeficiente de correlação de Spearman (SROCC), a raiz do erro médio quadrático (RMSE) e o rácio de *outliers* (OR) foram calculados entre os resultados subjetivos e os valores de avaliação de qualidade objetivos, aplicando um ajuste cúbico aos dados. Estes resultados foram calculados para avaliar linearidade, monotonia, precisão e consistência. Os resultados resumidos encontram-se na tabela 3.4

Tabela 3.4: Resultados de *benchmarking* das métricas objetivas tendo como referência os dados subjetivos obtidos neste estudo, com ajuste cúbico feito aos dados (Fonte: [22]).

	Métrica Objetiva	**PCC**	**SROCC**	**RMSE**	**OR**
UBI	p2point_MSE	0.549	0.725	1.130	0.917
	p2plane_MSE	0.504	0.640	1.169	0.889
	p2point_Haus	0.698	0.790	0.969	0.667
	p2plane_Haus	0.644	**0.800**	1.034	0.861
	PSNR-p2point_MSE	0.512	0.421	1.161	0.778
	PSNR-p2plane_MSE	0.439	0.341	1.215	0.861
	PSNR-p2point_Haus	0.646	0.607	1.032	0.833
	PSNR-p2plane_Haus	0.647	0.592	1.032	0.778
	MSE_YUV	**0.757**	0.760	**0.883**	**0.639**
	pl2plane_MSE	0.310	0.196	1.286	0.889
	PSNR	0.559	0.546	1.121	0.806
	SSIM	0.384	0.434	1.249	0.861
	MS-SSIM	0.571	0.619	1.110	0.861
	VIFP	0.753	0.767	0.890	0.694
UC	p2point_MSE	0.496	0.698	1.204	0.806
	p2plane_MSE	0.455	0.590	1.235	0.778
	p2point_Haus	0.632	0.786	1.075	0.722
	p2plane_Haus	0.566	**0.777**	1.143	0.861
	PSNR-p2point_MSE	0.480	0.378	1.216	0.750
	PSNR-p2plane_MSE	0.402	0.296	1.269	0.778
	PSNR-p2point_Haus	0.595	0.592	1.115	0.917
	PSNR-p2plane_Haus	0.590	0.559	1.119	0.806
	MSE_YUV	0.729	0.715	0.948	0.611
	pl2plane_MSE	0.308	0.238	1.319	0.833
	PSNR	0.547	0.523	1.160	0.806
	SSIM	0.356	0.425	1.296	0.861
	MS-SSIM	0.550	0.598	1.158	0.833
	VIFP	**0.731**	0.715	**0.946**	**0.583**
UNIN	p2point_MSE	0.525	0.723	1.163	0.778
	p2plane_MSE	0.486	0.614	1.194	0.750
	p2point_Haus	0.609	0.744	1.084	0.722
	p2plane_Haus	0.553	**0.760**	1.138	0.750
	PSNR-p2point_MSE	0.494	0.423	1.188	**0.694**
	PSNR-p2plane_MSE	0.413	0.324	1.244	0.861
	PSNR-p2point_Haus	0.602	0.572	1.092	0.778
	PSNR-p2plane_Haus	0.600	0.561	1.094	0.778
	MSE_YUV	0.731	0.714	0.932	**0.694**
	pl2plane_MSE	0.304	0.171	1.302	0.861
	PSNR	0.553	0.494	1.139	0.833
	SSIM	0.345	0.380	1.283	0.861
	MS-SSIM	0.538	0.567	1.152	0.833
	VIFP	**0.740**	0.723	**0.919**	0.722

É possível verificar que os resultados de correlação entre as métricas atuais de qualidade objetiva e os resultados subjetivos não são muito elevados o que leva à conclusão de que têm de ser encontradas novas formas objetivas de prever com exatidão a qualidade visual das PCs.

3.5 Teste suplementar de Avaliação Subjetiva de Qualidade de Nuvens de Pontos com Textura

Esta dissertação tem como ideia principal usar aprendizagem profunda para estimar a qualidade visual de PCs. Posto isto foi necessário realizar um novo estudo de avaliação subjetiva de qualidade de PCs para poder testar os algoritmos de aprendizagem profunda desenvolvidos nesta dissertação.

Este novo estudo foi realizado no Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Ciências e Tecnologias da Universidade de Coimbra.

3.5.1 Preparação dos conteúdos

Neste estudo foram utilizadas três PCs de pequena dimensão com informação de geometria e textura associadas. As PCs *Loot*, *Queen* e *Statue Klimt* fazem parte do *dataset* do projeto JPEG-PLENO [18], estas nuvens estão representadas na figura 3.12.

(a) *Loot*

(b) *Statue Klimt*

(c) *Queen*

Figura 3.12: Vista frontal de cada uma das PCs de referência.

Os algoritmos de compressão de PCs utilizados foram os mesmos utilizados em [17], *octree prunning* usando o *Point Cloud Library* e um codificador baseado em projeções implementado no *3DTK toolkit*.

Para cada um dos métodos foram usados os mesmos três níveis de qualidade da experiência anterior, um de qualidade baixa, um de qualidade média e outro de qualidade elevada, tal como descrito na secção 3.4. O número de pontos para cada conteúdo de referência e degradado, bem como outras informações como o tamanho do ponto definido estão listados na tabela 3.5.

Tabela 3.5: Configurações de compressão e renderização

Conteúdo	Tipo de compressão	Qualidade da compressão	# de pontos	% de pontos	Tamanho do ponto
Loot	**original**	–	805285	100	2
	3dtk	**Elevada**	799362	99.26	2
		Média	241207	29.95	3
		Baixa	63299	7.86	4
	pcl	**Elevada**	805285	100	2
		Média	241628	30.01	3
		Baixa	63122	7.84	4
Statue Klimt	**original**	–	499886	100	2
	3dtk	**Elevada**	483103	96.64	2
		Média	243718	48.75	3
		Baixa	71906	14.38	4
	pcl	**Elevada**	483094	96.64	2
		Média	244035	48.81	3
		Baixa	72316	14.47	4
Queen	**original**	–	1000993	100	2
	3dtk	**Elevada**	903410	98.24	2
		Média	252193	25.19	2
		Baixa	61604	6.15	4
	pcl	**Elevada**	1000993	100	2
		Média	252654	25.24	2
		Baixa	61735	6.16	4

Após a obtenção de todos os conteúdos foram gerados vídeos animados que foram avaliados com base na sua qualidade visual pelos participantes no estudo. Estes foram gerados com base num percurso de câmara, que consistiu na rotação da câmara à volta do eixo vertical e de seguida à volta do eixo horizontal com passos de 1° por *frame*. Para todos os casos a câmara foi colocada e movimentada de maneira a que os participantes conseguissem percecionar as PCs na sua totalidade. Este percurso resultou num total de 720 *frames* obtidos a partir do software *CloudCompare* [20], com resolução de 1920x1080 e a cor de fundo escolhida foi o preto. Estes foram comprimidos sem perdas com o codificador H.264/AVC(usando o *FFmpeg*), produzindo um vídeo animado a 30 *fps* com uma duração total de 24 *segundos* por conteúdo.

3.5.2 Metodologia de avaliação

As condições em que foram realizados os testes seguem a recomendação BT.500-13 do ITU-R [16]. Foi adotado o método *DSIS simultaneous* com uma escala de avaliação de 5 níveis (tabela 3.2), incluindo uma referência escondida para controlo.

Os vídeos da PC de referência e da PC degradada foram colocados lado a lado e cada observador teve de avaliar consoante o grau de similaridade com o conteúdo de referência. Foram obtidos um total de 18 resultados por avaliação em que cada observador avaliou três conteúdos diferentes com dois esquemas de compressão diferentes com três níveis distintos de qualidade, mais referências escondidas.

Os resultados obtidos foram filtrados para remover *outliers* tendo sido usado o método descrito na recomendação BT.500-13 do ITU-R [16], sendo que não foram encontrados nenhuns *outliers*. De seguida foram obtidos os *mean opinion scores* (MOS) e os intervalos de confiança de 95% assumindo uma distribuição T-Student's. Este estudo contou com um total de 20 pessoas, 10 do sexo feminino e 10 do sexo masculino com uma média de idades de 21,95 anos.

3.5.3 Resultados

O principal objetivo deste estudo é utilizar os resultados subjetivos obtidos para testar os algoritmos de aprendizagem profunda desenvolvidos nesta dissertação. Para fins de comparação é importante calcular as correlações com as métricas de avaliação de qualidade objetiva de nuvens de pontos existentes, bem como outros fatores de desempenho. Baseado na recomendação ITU-T P.1401 [21], o coeficiente de correlação de Pearson (PCC), o coeficiente de correlação de Spearman (SROCC), a raiz do erro médio quadrático (RMSE) e o rácio de outliers (OR) foram calculados entre os resultados subjetivos e os valores de avaliação de qualidade objetivos, aplicando um ajuste cúbico aos dados. Os resultados encontram-se na tabela 3.6

Tabela 3.6: Resultados de *benchmarking* das métricas objetivas tendo como referência os dados subjetivos obtidos neste estudo, com ajuste cúbico feito aos dados.

Métrica Objetiva	**PCC**	**SROCC**	**RMSE**	**OR**
p2point_MSE	0.9474	**0.8879**	0.4351	0.5
p2plane_MSE	0.9161	0.8755	0.8755	0.6111
p2point_Haus	0.9513	0.8071	0.4187	**0.444**
p2plane_Haus	**0.955**	0.8155	**0.403**	0.444
PSNR-p2point_MSE	0.8651	0.7845	0.6812	0.6111
PSNR-p2plane_MSE	0.7517	0.739	0.8957	0.6667
PSNR-p2point_Haus	0.9051	0.676	0.5775	0.5
PSNR-p2plane_Haus	0.8981	0.7039	0.5974	0.5556

Na generalidade os resultados de *benchmarking* das métricas objetivas deste estudo foram melhores do que os obtidos em [17], o que é de certa forma esperado uma vez que os conteúdos deste estudo são apenas três nuvens diferentes (ao contrário do que em [22] que conta com oito nuvens diferentes), comprimidas usando dois algoritmos distintos cada com três níveis de degradação associados.

4 Redes Neuronais Convolucionais

Redes Neuronais Convolucionais (Convolutional Neural Networks - CNNs) pertencem ao enorme universo do *Machine Learning* (ML), a ciência de programar computadores para que eles aprendam a partir de um conjunto de exemplos. Os dados exemplo são normalmente divididos em três partes:

- O conjunto de dados de treino (*dataset* de treino), que é o conjunto de dados que o algoritmo utiliza na sua tarefa de aprendizagem.
- O conjunto de dados de validação (*dataset* de validação) que serve essencialmente para testar o modelo com dados diferentes dos de treino durante o processo de treino. Isto faz com que o modelo ajuste de melhor forma aos dados e esteja melhor preparado para novos tipos de dados para além do *dataset* de treino.
- O conjunto de dados de teste (*dataset* de teste) que serve para testar a rede após o processo de treino, para ver se o modelo é capaz de prever de forma fiável valores para dados não usados no treino e validação.

Em geral, quanto maior for o conjunto de dados (*dataset* de treino + *dataset* de validação) maior a probabilidade de que a rede consiga fazer boas previsões para outros dados ainda não vistos (*dataset* de teste). O aumento do conjunto de dados também ajuda a prevenir o *overfitting*.

As Redes Neuronais Convolucionais (CNNs) surgiram a partir do estudo do córtex visual e têm sido utilizadas em reconhecimento de imagem desde os anos 80. Em 1981 os neurocientistas David H. Hubel e Torsten Wiesel mostraram que alguns neurónios do córtex visual têm um pequeno campo recetivo local, o que significa que apenas reagem a estímulos visuais localizados numa região limitada do campo visual. Os campos recetivos de diferentes neurónios podem-se sobrepor e juntos cobrem todo o campo visual. Os neurocientistas também provaram que alguns neurónios apenas reagem a padrões orientados na horizontal, enquanto outros reagem apenas a padrões com outras orientações. Também repararam que alguns neurónios têm campos recetivos maiores que outros e que podem reagir a padrões visuais mais complexos que são combinações de padrões de nível inferior. Isto levou à ideia de que neurónios de um nível superior se baseiam nas saídas dos neurónios de nível inferior. Uma arquitetura de processamento que emule este comportamento é capaz de detetar a maioria dos padrões visuais complexos em qualquer área do campo visual [2]. Estes estudos do córtex visual inspiraram a criação de CNNs. A arquitetura de uma CNN tem como base todos estes conceitos descritos anteriormente. CNNs usam como entrada direta imagens, e em vez de usar *features* definidas manualmente, as CNNs aprendem automaticamente uma hierarquia de *features* que é usada nas camadas seguintes. A construção das *features* é conseguida através de sucessivas convoluções da imagem de entrada com filtros, que constroem uma hierarquia de mapas de *features*. A abordagem hierárquica permite

à rede aprender *features* mais complexas em camadas superiores da rede [23]. A figura 4.1 exemplifica a estrutura de uma CNN.

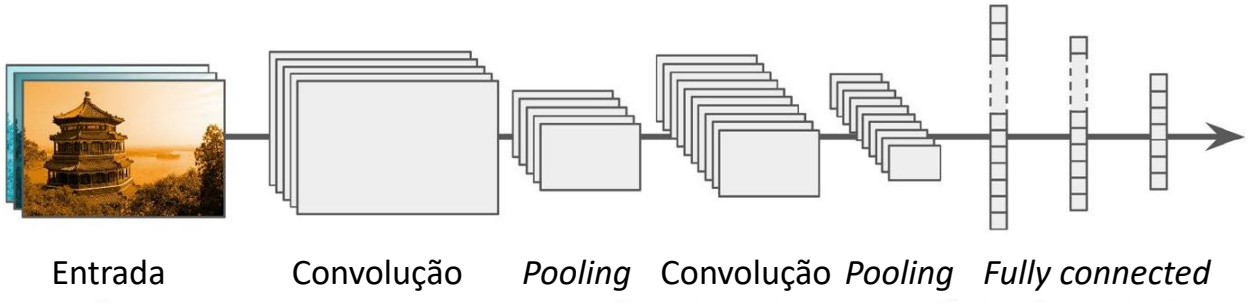


Figura 4.1: Exemplo de estrutura de uma CNN (Adaptada de [2]).

As CNNs têm apresentado desempenhos notáveis em tarefas de análise visual complexas. Elas são muito utilizadas em serviços de pesquisa de imagens, condução autónoma de veículos, classificação de vídeo automática e outros [2].

Mas nem tudo é bom neste tipo de redes e os algoritmos de ML têm alguns problemas conhecidos, como o *overfitting*, que significa que o modelo se ajusta perfeitamente aos dados de treino mas não generaliza bem, ou seja, apresenta piores resultados nos *dataset* de validação. Ora não é possível confiar numa rede com este tipo de problema.

Modelos complexos como CNNs podem detetar padrões subtis mas se o *dataset* de treino tiver demasiado ruído, ou se for demasiado pequeno então é provável que o modelo detete padrões do próprio ruído em vez de detetar padrões da informação útil [2].

Sistemas de ML, o que é o caso das CNNs, podem ser classificados de acordo com a quantidade e o tipo de supervisão que recebem durante o treino, existindo quatro categorias principais: aprendizagem supervisionada, aprendizagem não-supervisionada, aprendizagem semi-supervisionada e aprendizagem reforçada [2]. Nesta dissertação apenas irá ser abordada aprendizagem supervisionada.

4.1 Aprendizagem Supervisionada

Neste tipo de aprendizagem, os dados de treino fornecidos ao algoritmo incluem as soluções desejadas, chamadas de etiquetas (*labels*).

Um dos tipos de tarefas mais comuns em aprendizagem supervisionada é classificação. Para o treino deste tipo de tarefa a rede recebe conjuntos de imagens e classes respetivas [2]. Por exemplo para treinar uma rede para classificação de tipos de animais, na fase de treino a entrada da rede são as imagens dos animais e a sua respetiva classe (que tipo de animal se trata). Este algoritmo aprende a classificar os animais nas várias classes disponíveis.

Outra tarefa típica de aprendizagem supervisionada é regressão, e é utilizada quando é necessário que a rede tenha como saída um valor numérico alvo. Na fase de treino desta tarefa são fornecidas como entrada para além das imagens também o seu valor numérico correspondente sob forma de etiqueta, fazendo assim com que a rede aproxime um valor numérico a uma dada imagem com base nas caraterísticas da imagem.

4.2 Camadas constituintes de uma CNN

Como já foi ilustrada na figura 4.1 uma CNN é constituída por múltiplas camadas, sendo obrigatório uma camada de entrada e saída e algumas camadas intermédias, denominadas de camadas escondidas (*hidden layers*). Estas camadas escondidas podem ser divididas em vários tipos, camadas de ativação que são essencialmente funções transferência que se aplicam à saída dos nós para poder regularizar as saídas. Existem também as camadas de convolução que são as camadas mais importantes de uma CNN, para além destas há as camadas de pooling que têm como objetivo reduzir a carga computacional da rede e funcionam como um amostrador, e por fim existem camadas de normalização e de regularização que vieram resolver alguns problemas como por exemplo o *overfitting*. As próximas secções irão abordar com mais detalhe cada um destes tipos de camadas.

4.2.1 Camada Convolucional

O bloco mais caraterístico de uma CNN é a camada convolucional. Os neurónios da primeira camada convolucional não estão ligados a cada um dos pixéis da imagem de entrada, mas sim apenas aos pixéis que se encontram dentro dos seus campos recetivos, i.e na região dos dados de entrada que está a ser observada por uma determinada *feature*. Como ilustrado na figura 4.2.

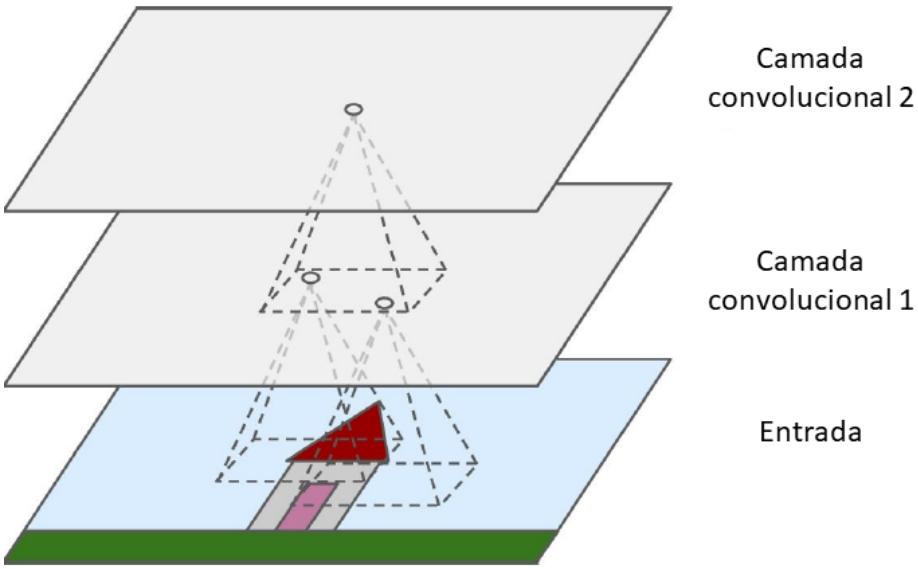


Figura 4.2: Camadas convolucionais da CNN com campos recetivos retangulares (Adaptada de [2]).

Por sua vez cada neurónio da segunda camada está ligado apenas a neurónios localizados dentro do retângulo na saída da primeira camada. Esta arquitetura permite que a rede se foque primeiro em *features* de baixo nível, da primeira camada, e que à posteriori as organize em *features* de nível superior da camada acima, e assim sucessivamente. Esta estrutura hierárquica é a razão pela qual as CNNs funcionam tão bem para reconhecimento de imagem [2].

Existem vários parâmetros que definem os campos recetivos e a sua deslocação ao longo da camada. Na figura 4.3 estão representados dois neurónios na camada superior (a vermelho e a azul) e na camada inferior os campos recetivos de onde originam.

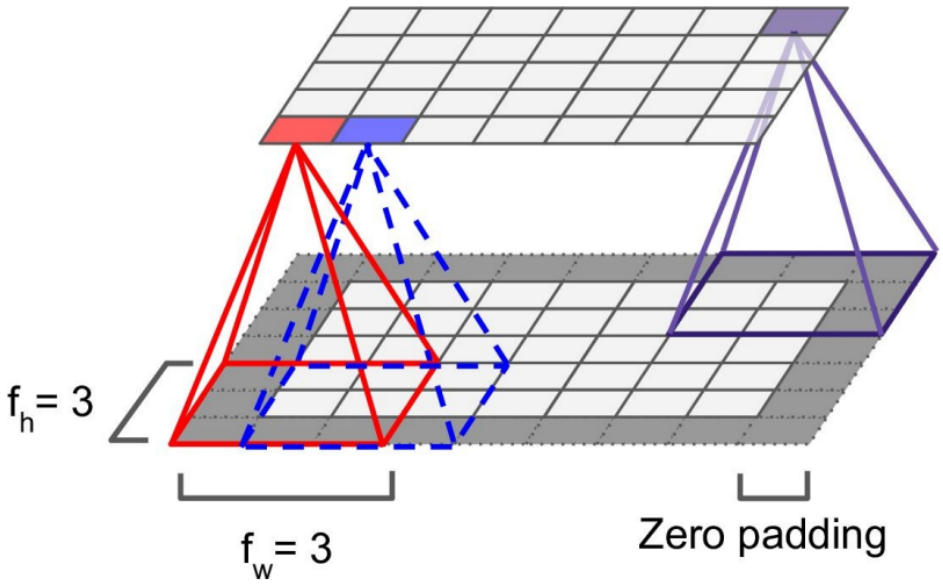


Figura 4.3: Ligações entre camadas (*stride* e *zero padding*) (Imagem de [2]).

Os primeiros parâmetros que caraterizam um campo recetivo são a sua largura e altura, no caso da figura cada campo recetivo tem altura de três píxeis e largura de três píxeis.

O parâmetro que define o deslocamento do campo recetivo é chamado de passo (*stride*) sendo que na figura 4.3 o passo é igual a um. O passo é aplicado tanto em largura como em altura na camada.

Aplicando um campo recetivo de 3x3 com um *stride* de um, a uma camada de resolução 5x7 obtém-se como saída uma camada de resolução 3x5. De forma a manter o tamanho da camada anterior na camada seguinte, o que é feito é acrescentar zeros nas margens da camada anterior como é ilustrado na figura 4.3 pelos píxeis mais escuros. Assim uma camada de 5x7 pode ter como saída uma camada de 5x7. Isto chama-se de extensão de sinal (*zero padding*) e é um procedimento que é definido pela quantidade de linhas de zeros extra que queremos acrescentar.

Filtros

Os pesos dados a cada neurónio não são mais que os coeficientes dos filtros ou *kernels* convolucionais. Durante o processo de treino os valores destes coeficientes são alterados por forma a otimizar o comportamento da rede.

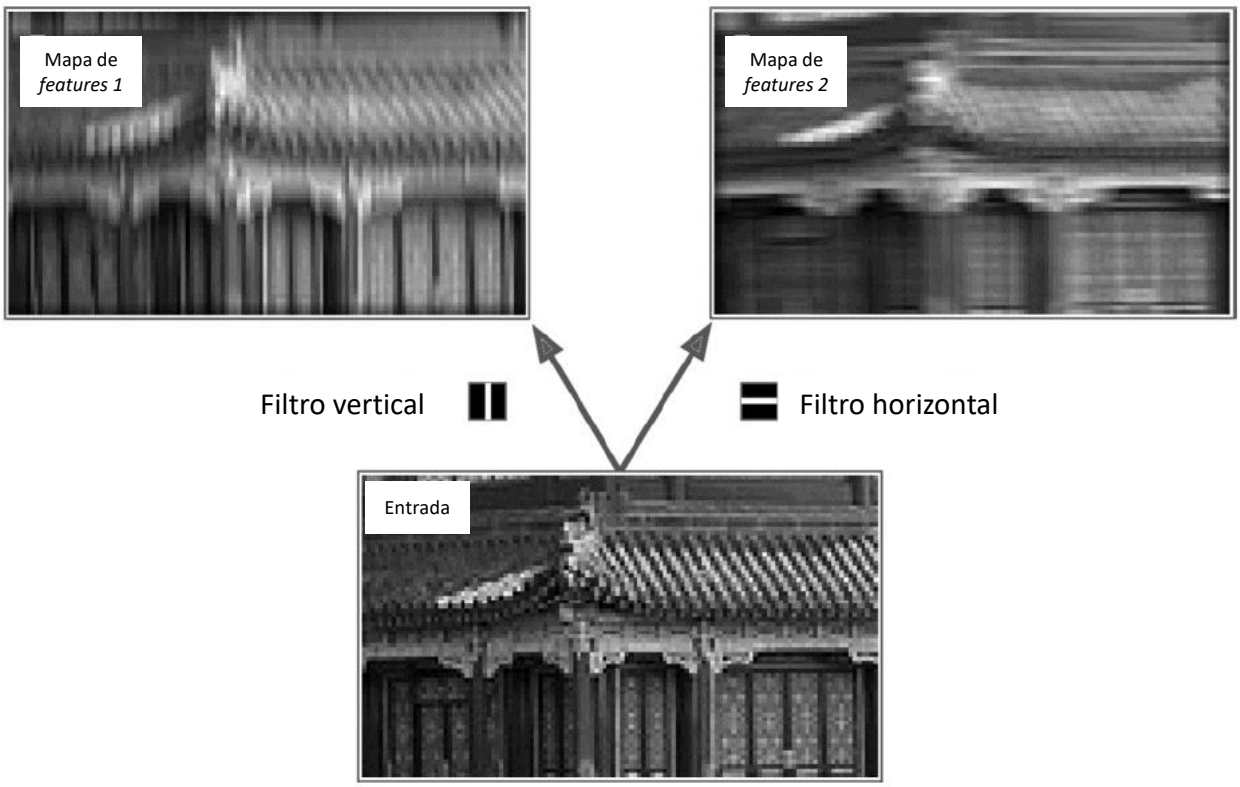


Figura 4.4: Aplicação de dois filtros diferentes (Adaptada de [2]).

Na figura 4.4 o filtro vertical está representado por um quadrado com uma linha branca vertical no meio, os neurónios que usarem estes pesos vão ignorar tudo no seu campo recetivo com exceção do conteúdo que estiver contido na linha branca vertical central. No filtro horizontal aplica-se exatamente o mesmo raciocínio mas neste caso a linha central é horizontal. Se a rede for alimentada pela entrada presente na figura 4.4 e se todos os neurónios de uma camada específica usarem o filtro vertical a saída irá ser a imagem no canto superior esquerdo da figura 4.4, onde é possível verificar que as linhas verticais da imagem foram realçadas. O mesmo acontece para o caso do filtro horizontal, que tem como saída o mapa de *features* localizado no canto superior direito da figura 4.4 em que foram realçadas as linhas horizontais da imagem.

Durante o treino os filtros da CNN são ajustados para que estes se adequem à tarefa pretendida sendo também aprendida a forma de os combinar para detetar padrões mais complexos [2].

Empilhamento de múltiplos mapas de *features*

Cada camada convolucional é composta por vários mapas de *features* de tamanhos iguais, sendo por isso mais conveniente representar as camadas em 3D como na figura 4.5.

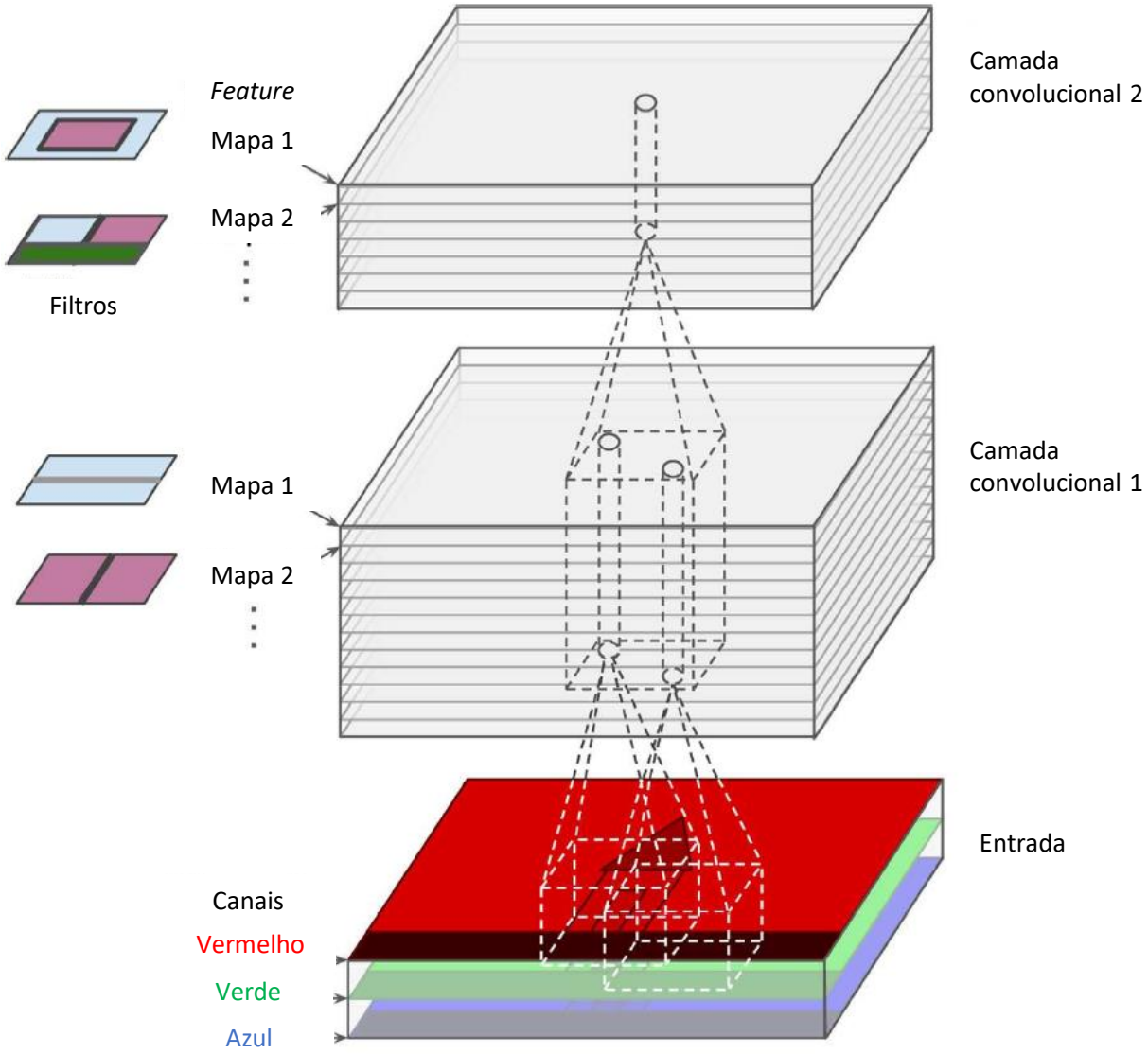


Figura 4.5: Representação 3D dos mapas de *features* (Adaptada de [2]).

Em cada um dos mapas de *features* todos os neurónios partilham dos mesmos parâmetros. Diferentes mapas podem ou não ter parâmetros diferentes.

Em resumo uma camada convolucional aplica em simultâneo múltiplos filtros às suas entradas fazendo com que sejam capazes de detetar múltiplas *features* em qualquer parte das suas entradas [2].

4.2.2 Camada *fully connected*

Esta camada implementa a parte de classificação ou regressão, sendo de facto o classificador da CNN. Recebe como entrada as *features* calculadas no estágio convolucional e usando uma das várias arquiteturas possíveis para classificadores calcula a regressão ou classificação final.

4.2.3 Camada de Retificação Linear (ReLU)

Esta camada é utilizada como função de ativação em modelos de aprendizagem profunda. Esta função tem como saída zero se receber como entrada valores negativos e tem como saída o valor de entrada se este for positivo. É graficamente representada como na figura 4.6.

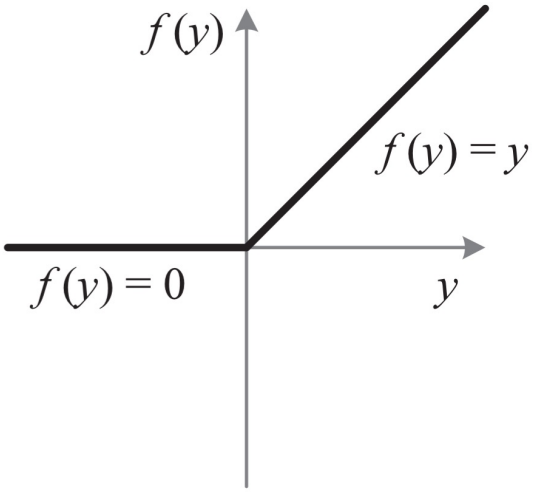


Figura 4.6: Gráfico da função de retificação linear.

Aplicando esta função à saída de uma transformação linear obtém-se uma transformação não-linear, no entanto a função permanece muito próxima da linearidade sendo que se trata de uma função linear por partes. Por isto esta função mantém muitas das propriedades que fazem com que os modelos lineares sejam mais fáceis de otimizar com métodos baseados em gradientes [24].

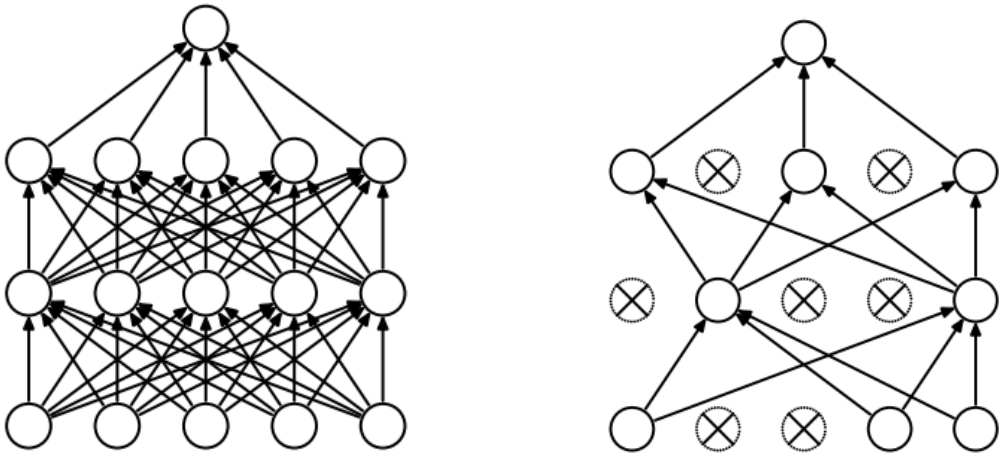
4.2.4 Camada de *Pooling*

O objetivo da camada de *pooling* é sub-amostrar os mapas de saídas das várias camadas convolucionais de maneira a reduzir a carga computacional, o uso de memória e o número de parâmetros.

Tal como na camada convolucional cada neurónio na camada de *pooling* é ligado às saídas de um número limitado de neurónios da camada anterior (campo recetivo). No entanto um neurónio de *pooling* não tem pesos, tudo o que faz é juntar as entradas usando uma função como por exemplo o máximo ou a média. Numa camada de *pooling* máximo apenas a entrada de valor máximo em cada *kernel* passa para a camada seguinte [2].

4.2.5 *Dropout*

A técnica de *dropout* é uma técnica de regularização que previne o *overfitting* e possibilita a combinação exponencial de diferentes arquiteturas numa CNN. *Dropout* significa excluir temporariamente da rede uma unidade assim como as suas ligações à rede, como demonstra a figura 4.7. A escolha dos elementos que vão sofrer *dropout* é aleatória. No caso mais simples para cada unidade é fixa uma probabilidade de retenção p independente dos outros elementos, onde p pode ser escolhida usando um conjunto de validação ou simplesmente definida para 0.5, que é próximo do valor ótimo para a maior parte das redes e de tarefas [25].



(a) Rede Normal

(b) Rede depois de *dropout*

Figura 4.7: Exemplo *dropout* em redes neuronais (Adaptada de [25])

Aplicar *dropout* a uma rede é o mesmo que amostrar a rede em várias redes mais simples, tal com está presente na figura 4.7b. Uma rede com n elementos pode ser vista como possíveis 2^n redes mais simples.

Os autores de [25] descobriram que o uso de *dropout* durante o treino de uma rede neuronal leva a um decréscimo significativo do erro para uma grande variedade de tarefas de classificação, comparando com outros métodos de regularização.

4.3 Hiper-Parâmetros do Algoritmo de Treino

Os Hiper-Parâmetros são variáveis que são definidas antes de iniciar o processo de aprendizagem que determinam como a rede é treinada.

4.3.1 Taxa de aprendizagem

Um dos hiper-parâmetros mais importantes é a taxa de aprendizagem e define a velocidade com que o sistema se adapta aos diferentes dados. Se for definida uma taxa de aprendizagem muito alta então o sistema vai adaptar-se muito rapidamente aos novos dados mas também vai tender a esquecer os dados antigos mais rapidamente. Por outro lado se a taxa de aprendizagem for muito baixa o sistema irá ter mais inércia à adaptação a novos dados o que significa que irá aprender mais lentamente, mas também irá ser menos sensível a ruído dos novos dados.

4.3.2 Tamanho do *Minibatch*

O tamanho do *minibatch* é um hiper-parâmetro que define o conjunto de amostras processadas pela rede entre atualizações dos seus parâmetros internos. A rede percorre todas as amostras definidas pelo tamanho do *minibatch* e faz as previsões para esse conjunto de amostras. No fim as previsões são comparadas com os valores de saída esperados e é calculado o erro. A partir deste valor de erro os coeficientes da rede são ajustados de maneira a melhorar as próximas previsões. E assim sucessivamente.

4.3.3 Número de épocas

O número de épocas é o hiper-parâmetro que define o número de vezes que o algoritmo de aprendizagem percorre todo *dataset*. Apenas vale a pena haver mais épocas se o erro de validação continuar a decrescer. Assim que o erro de validação estagnar ou aumentar o algoritmo deverá ser interrompido.

4.4 Exemplos de Redes Neuronais Convolucionais

Nos últimos anos foram propostas e testadas várias arquiteturas de redes neuronais convolucionais, algumas delas com resultados impressionantes principalmente em problemas de classificação. Em competições como por exemplo a *ImageNet Large Scale Visual Recognition Competition (ILSVRC)*, a taxa de erros do top-5 de redes para classificação de imagens caiu de 26% para 3% em apenas 5 anos. A taxa de erros do top-5 é a percentagem de imagens de teste para qual o sistema em nenhuma das cinco previsões de probabilidade mais elevada acertou na classe correta [2].

Nesta secção irão ser descritas as arquiteturas das CNNs que apresentaram melhores resultados na ILSVRC entre 2012 e 2015. Foram estas três redes que foram usadas nesta dissertação com *transfer learning* que consiste em encontrar redes pré-treinadas utilizadas para tarefas semelhantes e reutilizá-las, afinando-as para o problema que se pretende resolver. Isto consegue-se reutilizando algumas camadas previamente treinadas, o que acelera consideravelmente o treino e requer muito menos dados de treino.

4.4.1 AlexNet

Uma das primeiras redes desenvolvidas que ganhou o *ILSVRC ImageNet challenge* em 2012 com um resultado de taxa de erros top-5 de 17% foi a *AlexNet CNN architecture* [26], foi desenvolvida por Alex Krizhevsky, Ilya Sutskever, e Geoffrey Hinton. Esta rede segue a arquitetura documentada na tabela 4.1.

Tabela 4.1: Arquitetura *AlexNet* (Adaptada de [2]).

Camada	Tipo	Mapas	Tamanho	Tamanho dos filtros	Passo	*Padding*	Ativação
Out	Fully Connected	–	1,000				Softmax
F9	Fully Connected	–	4,096				ReLU
F8	Fully Connected	–	4,096				ReLU
C7	Convolução	256	13 × 13	3 × 3	1	MESMO	ReLU
C6	Convolução	384	13 × 13	3 × 3	1	MESMO	ReLU
C5	Convolução	384	13 × 13	3 × 3	1	MESMO	ReLU
S4	*Pooling* Máximo	256	13 × 13	3 × 3	2		
C3	Convolução	256	27 × 27	5 × 5	1	MESMO	ReLU
S2	*Pooling* Máximo	96	27 × 27	3 × 3	2		
C1	Convolução	96	55 × 55	11 × 11	4	MESMO	ReLU
In	Entrada	3 (RGB)	224 × 224				

Para reduzir o *overfitting* foram usadas duas técnicas de regularização, foi aplicado *dropout* nas saídas das camadas F8 e F9, e foi usada a técnica de *data augmentation*. Para além disto a *AlexNet* também aplica um passo de normalização a seguir à *ReLU* nas camadas C1 e C3, chamada de normalização de resposta local.

4.4.2 GoogLeNet

A *GoogLeNet* foi desenvolvida por Christian Szegedy et al. da *Google Research* [27] e ganhou o *ILSVRC ImageNet challenge* de 2014 conseguindo um resultado top-5 de taxa de erros de 7%. Isto aconteceu devido ao facto desta rede ser muito mais profunda que as até então testadas, através de sub-redes chamadas de *inception modules* que permitem que a *GoogLeNet* utilize os parâmetros de uma forma muito mais eficiente [2].

Cada um destes módulos segue a arquitetura da figura 4.8 em que "1X1"é o tamanho do *kernel* e "+1(S)"significa um *stride* de 1 [2]. É percetível que o sinal de entrada serve de entrada de cada ramo, mas cada um destes ramos tem camadas diferentes , o que permite capturar padrões a escalas diferentes que no fim são concatenados na última camada do módulo.

Cada módulo destes funciona como uma camada convolucional mas com a capacidade de produzir mapas de *features* capazes de percecionar padrões mais complexos a diferentes escalas.

A *GoogLeNet* segue a arquitetura da figura 4.9 em que as caixas com os piões são os *inception modules* mencionados anteriormente.

As primeiras duas camadas dividem a imagem em 16 partes iguais de forma a reduzir o custo compu-

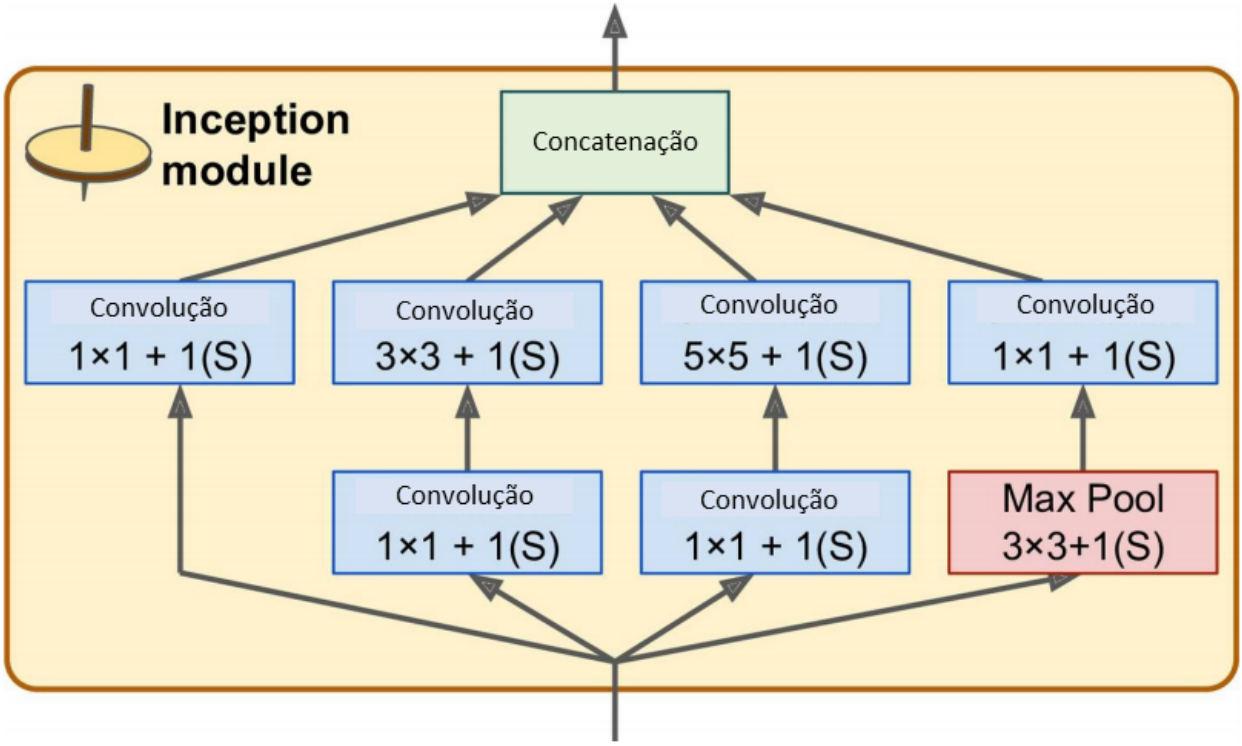


Figura 4.8: *Inception module* da *GoogLeNet* (Adaptada de [2]).

tacional. De seguida a camada de normalização garante que as camadas anteriores aprendem uma grande variedade de *features*. Seguem-se duas camadas convolucionais que podem ser abordadas como uma só camada convolucional que produz mapas de *features* capazes de percecionar padrões mais complexos. É feita de novo uma normalização de resposta local seguida de uma camada de *pooling* máximo que reduz a altura e a largura da imagem para metade para acelerar os cálculos. As próximas camadas dizem respeito à pilha de *inception modules* alternados com camadas de *pooling* máximo, mais uma vez para acelerar os cálculos. No fim dos *inception modules* vem uma camada de *pooling* médio que usa um *kernel* do tamanho dos mapas de *features* com *padding* válido, tendo como saída mapas de *features* de tamanho 1x1. Esta estratégia tem o nome de *pooling* de média global, e evita a necessidade de ter várias camadas *fully connected* no final da rede como acontecia na *AlexNet*, reduzindo assim o número de parâmetros da rede e limitando o risco de *overfitting*. Finalmente as últimas camadas, uma de *dropout* para regularização e outra *fully connected* com ativação *softmax* para obter como saída as probabilidades de classe estimadas [2].

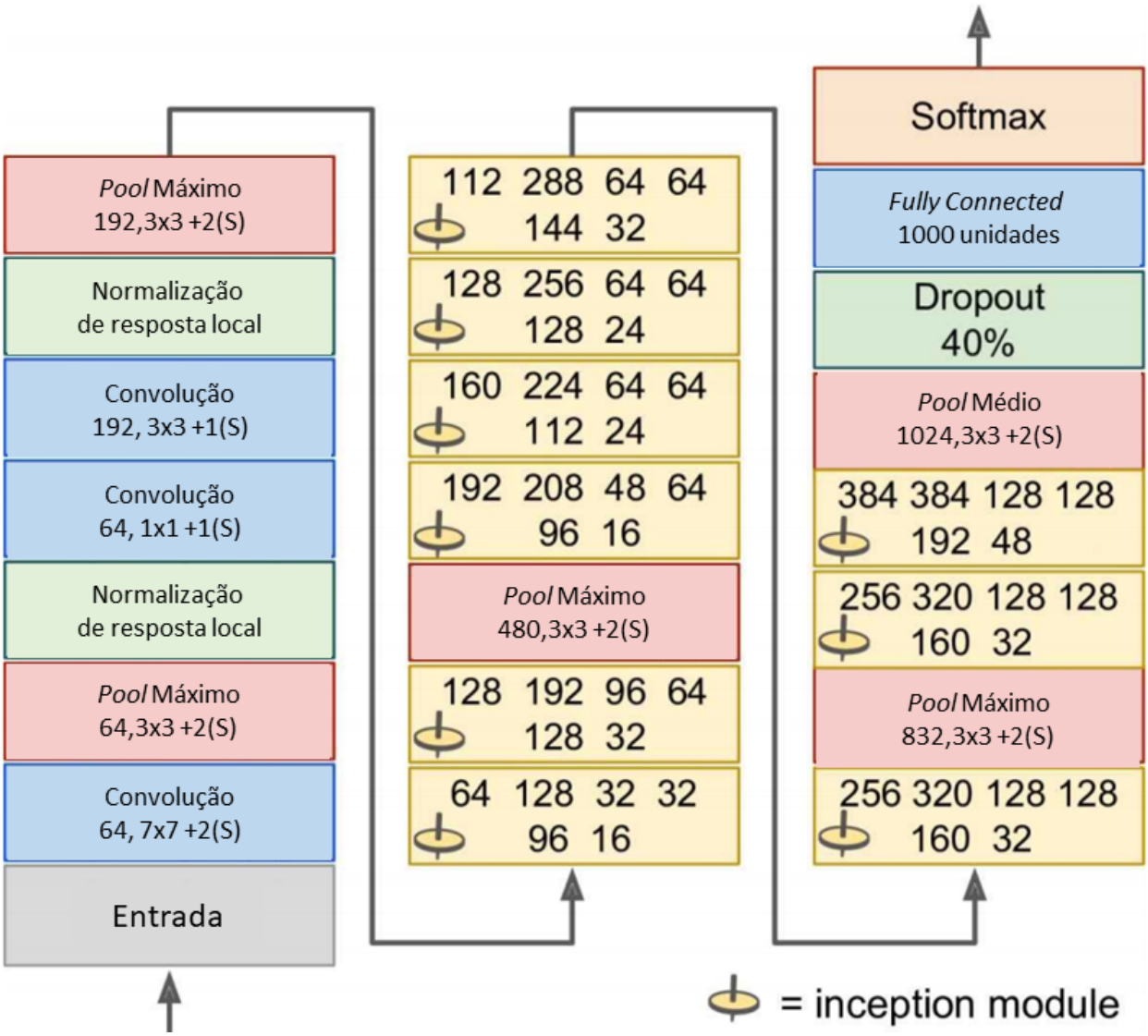


Figura 4.9: Arquitetura da *GoogLeNet* (Adaptada de [2]).

4.4.3 ResNet

Em 2015 a vencedora do *ILSVRC ImageNet challenge* foi a *Residual Network* (*ResNeT*) desenvolvida por Kaiming He et al. [28] conseguindo um resultado top-5 de taxa de erros abaixo de 3,6% sendo uma CNN extremamente profunda com 152 camadas. Para poder treinar uma rede tão profunda têm de ser usadas conexões de atalho. Estas conexões fazem com que na saída de uma determinada ou determinadas camadas seja adicionada a entrada das mesmas, tal como demonstrado na figura 4.10 [2].

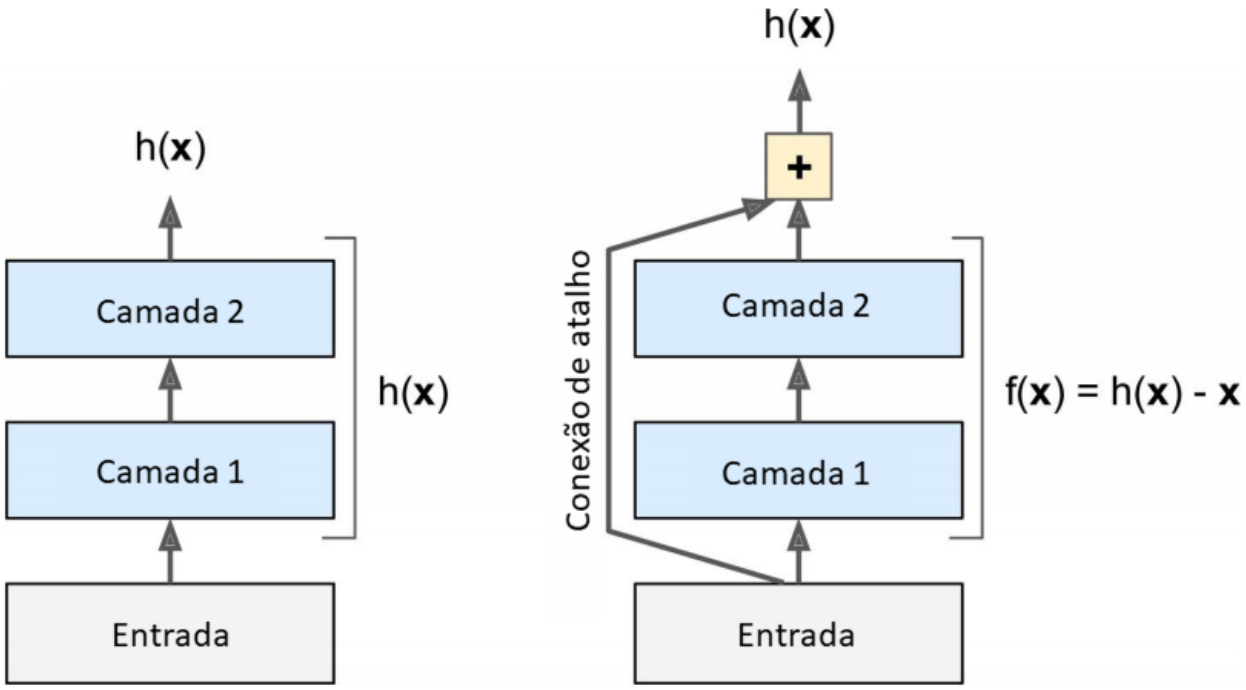


Figura 4.10: *Residual Learning* (Adaptada de [2])

No treino de uma rede neuronal o objetivo é que as camadas modelem uma função alvo **h(x)**, se for adicionada a entrada **x** à saída (conexão de atalho) então a rede iria ser forçada a modelizar **h(x)-x** em vez de simplesmente **h(x)**. Isto é chamado de *residual learning*.

Quando uma rede é inicializada os seus pesos são próximos de zero então a rede apenas tem como saída valores próximos de zero. Se forem adicionadas conexões de atalho, a saída da rede em vez ter valores próximos de zero irá ter como saída uma cópia da entrada. O que significa que a rede tende a modelar a função identidade, o que, se a função alvo for próxima da identidade (que é frequente), acelera consideravelmente a fase de treino. Graças às conexões de atalho o sinal pode facilmente percorrer a rede toda.

A arquitetura da *ResNet*, presente na figura 4.11, começa e acaba de forma exatamente igual à *GoogleNet*, no meio tem uma pilha muito profunda de unidades residuais. Cada unidade residual é composta por duas camadas convolucionais com normalização de *batch* e ativação *ReLU*, que usam *kernels* de 3x3 com *stride* e *padding* de uma unidade [2].

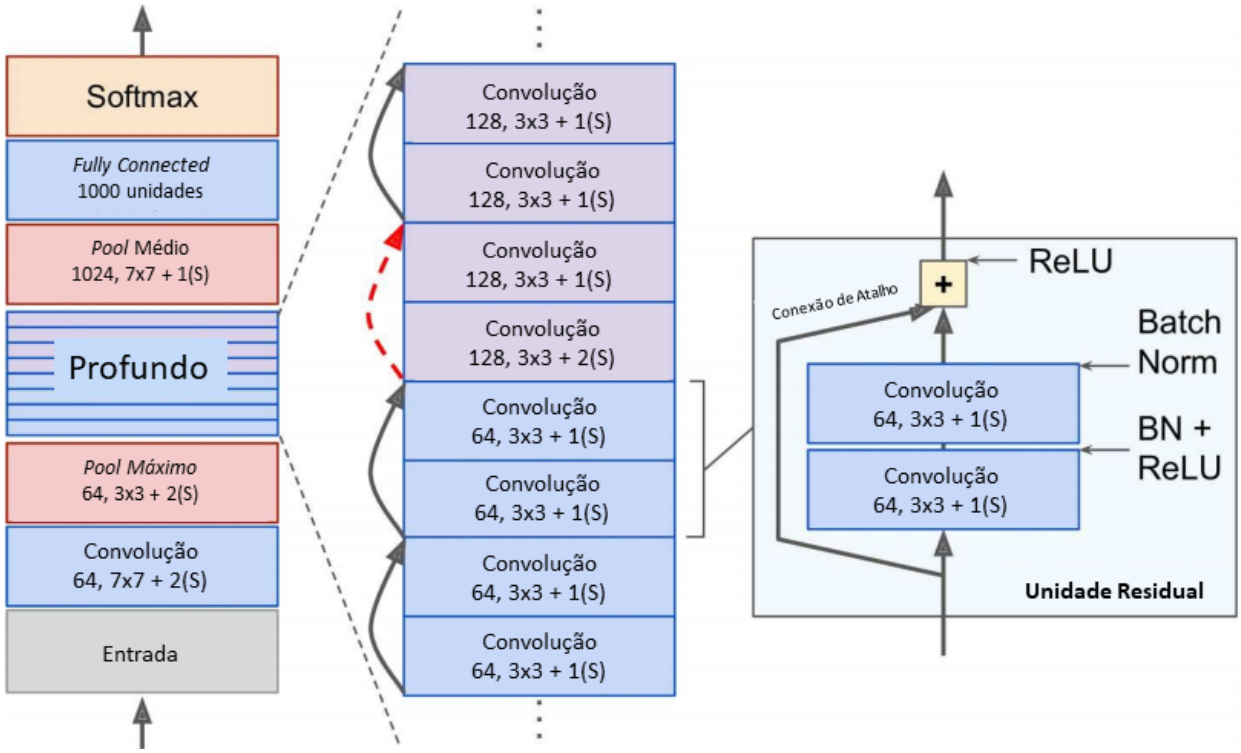


Figura 4.11: Arquitetura *ResNet* (Adaptada de [2])

Em algumas unidades residuais o número de mapas de *features* é o dobro, ao mesmo tempo que a sua altura e largura são reduzidas para metade usando uma camada convolucional com *stride* de 2. Quando isto acontece as entradas não podem ser adicionadas diretamente à saída pois estas não têm o mesmo tamanho. Esta conexão de atalho está assinalada na figura 4.11 com a seta a tracejado vermelho. Para resolver este problema em vez de a conexão de atalho ser uma ligação direta da entrada à saída, a entrada passa por uma camada convolucional de de 1x1 e com *stride* de dois e com o número certo de mapas de *features* [2]. Isto faz com a que entrada tenha as mesmas dimensões que a saída.

Existem várias versões da *ResNet* com vários níveis de profundidade, a *ResNet-50*, *ResNet-101* e *ResNet-152*. Essencialmente o que varia entre elas é o nível de profundidade, sendo que o nível de profundidade é indicado pelo número que lhe sucede.

4.5 Redes Neuronais Convolucionais usadas para Estimação de Qualidade

Há bem pouco tempo problemas de visão por computador, tais como classificação de imagens e deteção de objetos eram abordados em dois passos: Construção manual de *features* específicas para o problema em questão, e construção de algoritmos de aprendizagem para regressão ou classificação. Ao longo dos últimos anos as CNNs mostraram superar estas técnicas tradicionais, também em aplicações de avaliação de qualidade de imagem e vídeo.

Em [3] foi desenvolvido um trabalho que utiliza uma CNN para avaliação de qualidade de imagem NR 2D. Esta CNN utiliza como entrada pequenos pedaços de uma imagem (*patches*). A estrutura da rede consiste numa camada convolucional com 50 mapas de *features* associada a uma camada de *pooling* máximo e uma

camada de *pooling* mínimo seguida de duas camadas *fully connected* e um nó de saída, tal como ilustra a figura 4.12.

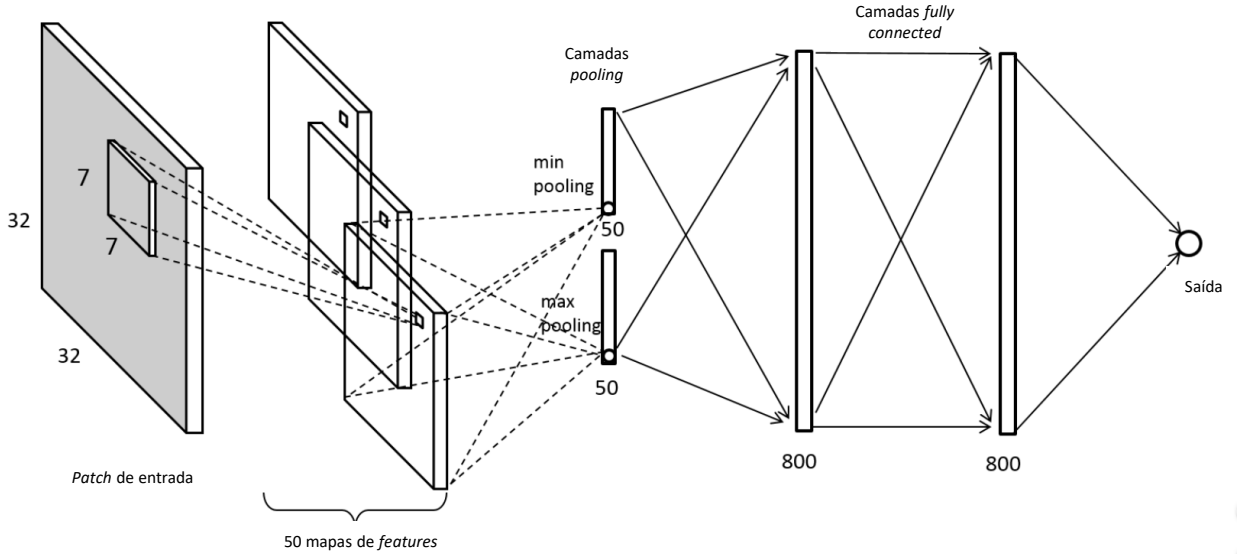


Figura 4.12: Arquitetura da CNN (Adaptada de [3]).

Esta abordagem NR com uma CNN atinge melhores resultados de correlação com resultados de avaliação subjetiva de qualidade do que as métricas de qualidade objetiva disponíveis na altura, entre elas PSNR, SSIM e FSIM [3].

Em [4] é proposta a utilização de uma CNN para avaliação objetiva de qualidade FR. Esta CNN é composta por dez camadas convolucionais, cinco camadas de *pooling* máximo e duas camadas *fully connected* e um nó de saída. A entrada é um *patch* da imagem de referência e outro *patch* da imagem degradada. Foi usada uma rede siamesa (*siamese network*) para aprender as relações de similaridade das entradas. Para isto os *patches* de entrada são processados em paralelo pelas duas redes siamesas que partilham os pesos das suas ligações sinápticas. Estas redes são usadas para extração de *features*. Para usar estas *features* no problema de regressão é necessário fundir as *features*. Por sua vez estas *features* fundidas são a entrada para a parte de regressão da rede. Esta arquitetura está ilustrada na figura 4.13.

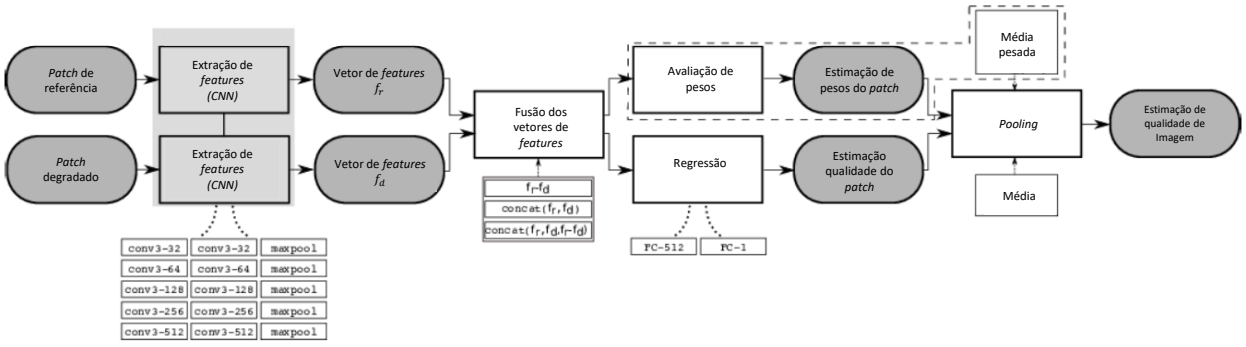


Figura 4.13: Arquitetura da CNN (Adaptada de [4]).

Os resultados destes métodos com utilização de CNNs para avaliação de qualidade objetiva FR superam os outros resultados de estado da arte. O principal problema destes métodos é a necessidade de dispor de muitos dados para treino [4].

5 Avaliação de Qualidade de Nuvens de Pontos com Redes Neuronais Convolucionais

Após toda a revisão feita no capítulo 3, foi possível concluir que existe muito trabalho para ser desenvolvido no que diz respeito a avaliação de qualidade de Nuvens de Pontos, com métodos computacionais capazes de estimar corretamente a qualidade subjetiva.

Os estudos [22] e [17] vieram confirmar que existe uma grande discrepância entre os resultados subjetivos e as métricas de qualidade objetivas. O que, assumindo os resultados subjetivos como valor de referência, indica que as métricas de qualidade objetiva atuais não podem ser consideradas bons preditores da qualidade visual das PCs.

Este trabalho tem como objetivo principal construir uma nova forma de avaliar a qualidade visual de nuvens de pontos, tendo sempre os resultados subjetivos de [17] como resultados de referência. A maior parte do trabalho feito em qualidade objetiva de PCs baseia-se apenas na geometria da PC e nas relações entre os pontos. Isto pode não ser a melhor solução, uma vez que em [17] estas métricas baseadas em geometria não obtiveram boas correlações com resultados de avaliação subjetiva .

Tal como referido na secção 3.2.4, os autores de [14] propuseram uma métrica que avalia a qualidade de uma PC com base nas suas projeções sobre as várias faces de um cubo que inscreve toda a PC. As projeções são de imagens em 2D, o que leva a que seja possível implementar métricas de avaliação de qualidade visual de imagens 2D para estimação de qualidade de PCs.

A ideia de que uma nuvem de pontos pode ser representada por múltiplas projeções foi o ponto de partida deste trabalho. Este tipo de representação tem múltiplas vantagens. Uma delas é poder aplicar técnicas já conhecidas para imagem 2D a nuvens de pontos, o que facilita a abordagem ao problema de avaliação de qualidade visual das PCs. No entanto existem desvantagens, uma delas é o facto das projeções não representarem a nuvem de pontos na sua totalidade. Outra é atenuar as características imersivas deste tipo de conteúdo.

A secção 4.5 mostra que já foram implementadas CNNs para avaliação objetiva de qualidade de imagem 2D com resultados que superam os resultados de métricas de estado da arte.

A ideia deste trabalho é usar as projeções 2D nas faces de um cubo que inclui toda a PC, de várias PCs juntamente com os resultados subjetivos obtidos em [17] para afinar as várias CNNs pré-treinadas referidas na secção 4.4 usando o método de *transfer learning*, com o objetivo de avaliar objetivamente a qualidade visual de uma PC, para que, desta forma se obtenham resultados objetivos de avaliação de qualidade que apresentem maior correlação com os resultados subjetivos. O diagrama deste processo está ilustrado na Figura 5.1

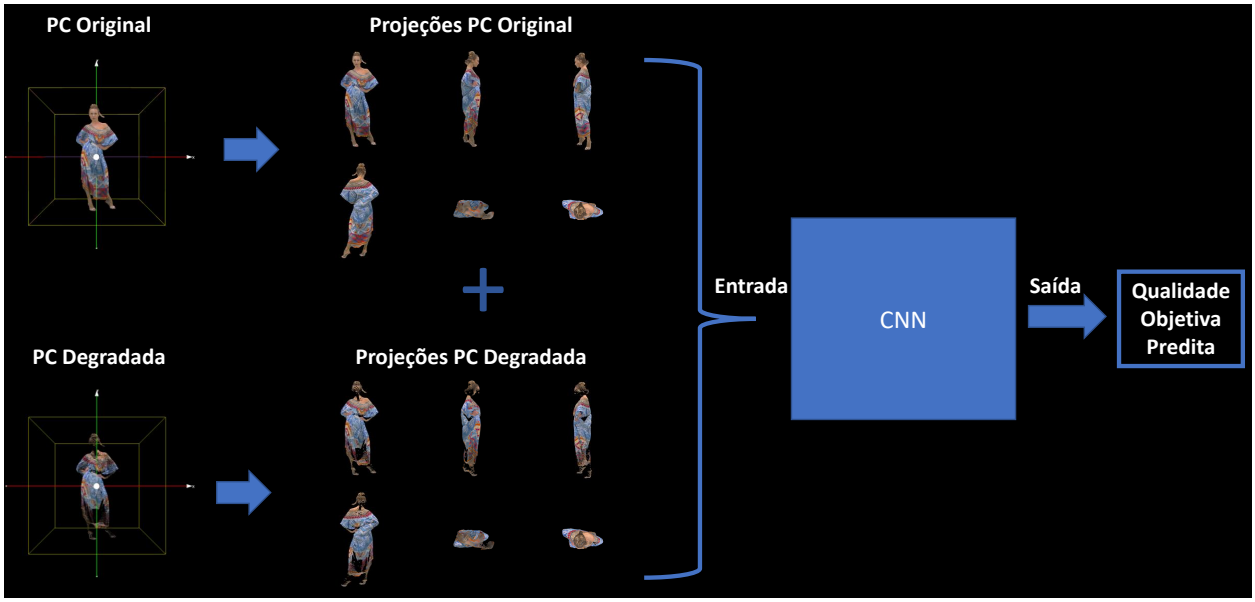


Figura 5.1: Diagrama do processo de estimação de qualidade objetiva recorrendo a CNNs.

5.1 *Dataset Principal*

Tal como referido no capítulo 4 quanto maior o *dataset* maior a probabilidade de sucesso no treino da rede, pelo que a preparação do *dataset* é uma das tarefas mais importantes no processo de treino de uma CNN.

Um dos problemas deste tipo de conteúdo é, pela sua novidade, ainda não terem sido realizados muitos estudos de avaliação subjetiva de nuvens de pontos. Como a ideia principal é obter uma medida que vá ao encontro dos resultados subjetivos isto restringe o *dataset* ao estudo disponível em [17].

Todos os estudos abordados na secção 4.5 usaram como *dataset* imagens 2D e obtiveram bons resultados. Para além disto todas as redes descritas na secção 4.4 têm como entrada o mesmo tipo de imagens. Por estas razões optou-se por utilizar as projeções de nuvens de pontos com textura. O único estudo disponível de avaliação de qualidade subjetiva de PCs com textura é o estudo [17], pelo que o *dataset* foi contruído com essas PCs. Dessas PCs apenas foram utilizadas as nuvens de pequena escala, *longdress*, *romanoillamp* e *bumbameuboi*, originais e com vários níveis de degradação.

5.1.1 Extração de projeções das Nuvens de Pontos

Regra geral uma CNN tem como entrada imagens 2D, para isso foi necessário obter as projeções das PCs nas faces de um cubo que inclui toda a PC. Isto foi conseguido recorrendo às ferramentas de renderização disponíveis na biblioteca PCL [7]. Esta extração de projeções está ilustrada na figura 5.2 onde se mostram a PC e seis projeções 2D sobre as faces de um cubo que envolve toda a PC.

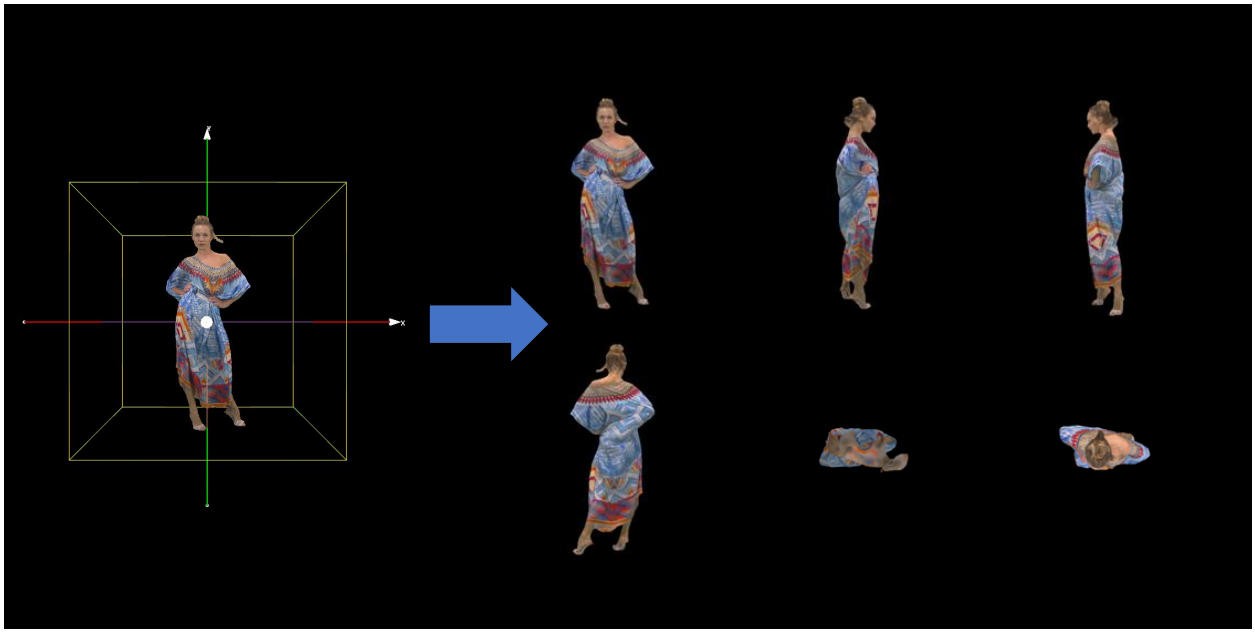
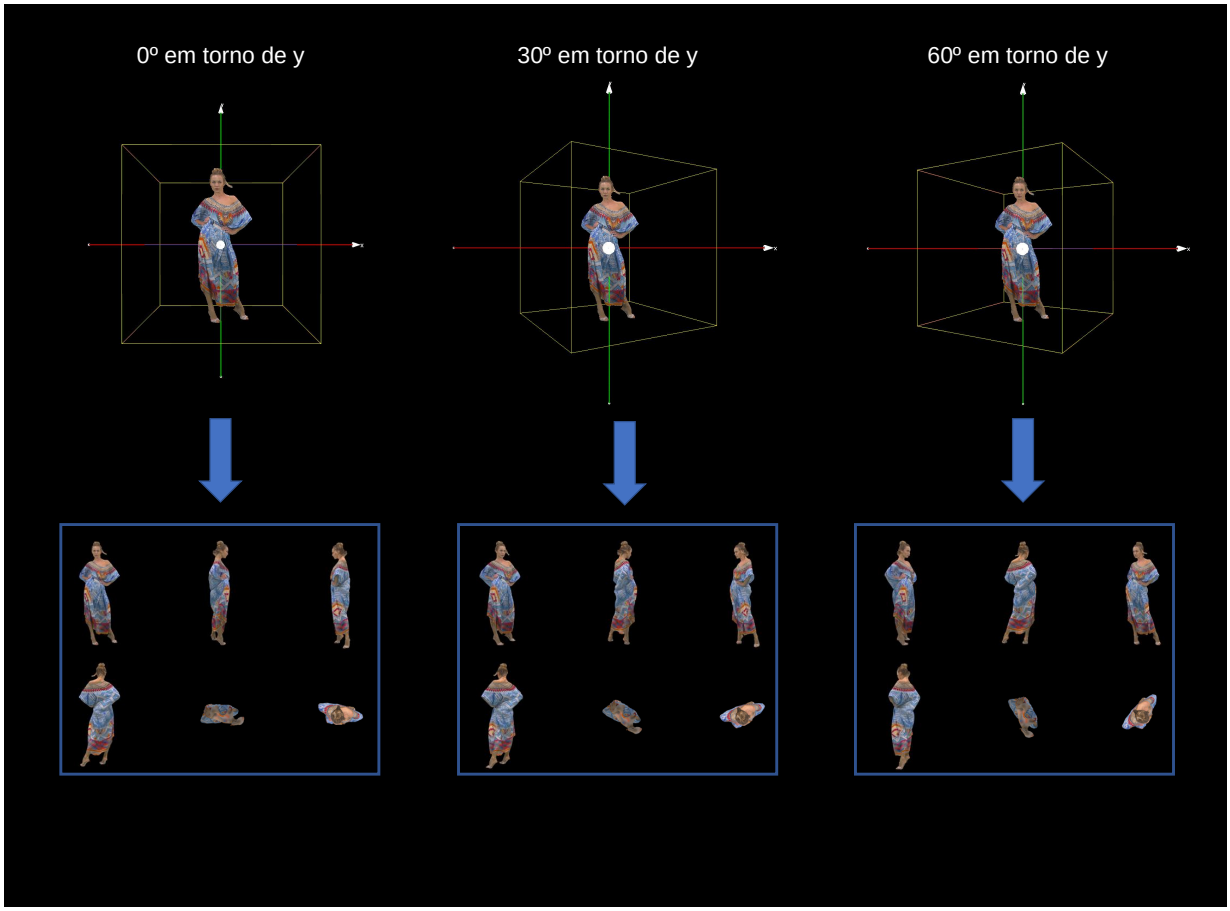
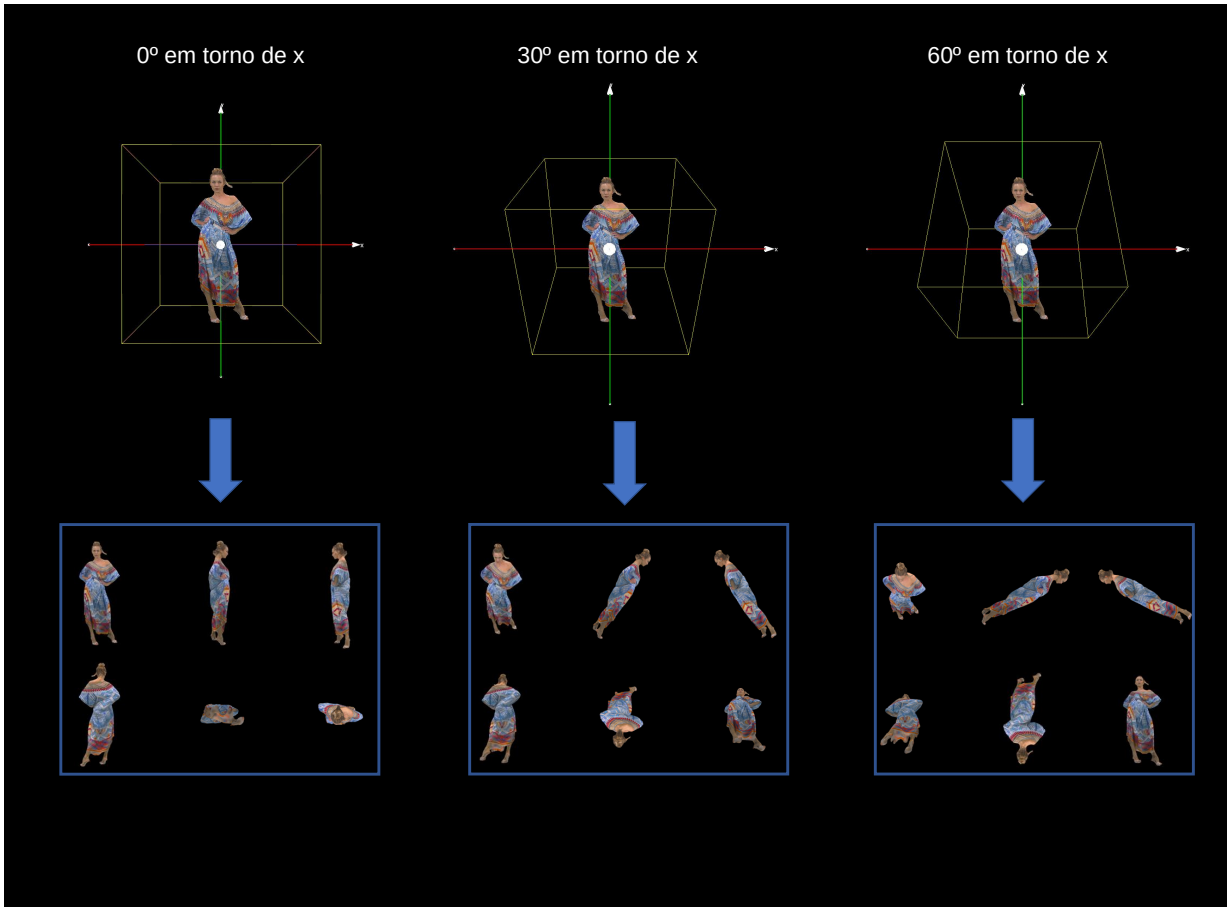


Figura 5.2: Extração de projeções na *bounding box* da PC *longdress*

Não é possível caracterizar completamente uma PC recorrendo somente às suas projeções. Tendo isto em consideração e a necessidade de mais dados de entrada para a CNN, para cada PC o cubo foi rodado de 1° em 1°, primeiro em torno do eixo dos *yys* e depois em torno do eixo dos *xxs*, e para cada rotação de 1° tiradas as suas projeções, tal como ilustra o esquema da figura 5.3. Para cada rotação de 1° foram obtidas as projeções da PC nas seis faces do cubo tal como representado na figura 5.3 com os exemplos de 0°, 30° e 60°, isto em torno de y na figura 5.3a e em torno de x como na figura 5.3b.



(a) Rotação em torno de y.



(b) Rotação em torno de x.

Figura 5.3: Exemplos de extração de projeções com rotação do cubo com 0°, 30° e 60°.

Cada projeção foi obtida com uma resolução de 1920x1080 píxeis. Cada uma destas imagens tem uma grande margem de fundo preto, pelo que foi necessário verificar a sua área de importância para definir as fronteiras mínimas de cima de baixo e dos lados. Definindo estas fronteiras foi possível cortar as imagens todas com o mesmo tamanho. Este corte das margens de fundo preto resultou numa redução da resolução das imagens para 729x729 píxeis. Apesar da redução significativa na resolução das imagens ainda foi preciso fazer um *resize* das imagens de 729x729 píxeis para 224x224 píxeis por forma a coincidirem com o tamanho das imagens de entrada das CNNs referidas na secção 4.4.

Esta rotação de 1° do cubo permitiu aumentar consideravelmente os dados de entrada da CNN, pois para cada PC temos 360 conjuntos de projeções com rotação em torno de x, mais 360 conjuntos de projeções com rotação em torno de y. Isto também permitiu fornecer à CNN todos os *frames* visualizados pelos participantes do estudo [17].

Este processo foi repetido para todas as PCs de pequena escala usadas em [17], *longdress*, *romanoillamp* e *bumbameuboi*, originais e com três níveis de degradação, comprimidas por dois softwares diferentes (*3dtk* e *pcl*), tal como descrito na secção 3.4.

5.1.2 Organização do *Dataset*

O desafio de utilizar uma CNN como um estimador FR de qualidade objetiva é saber que dados de entrada fornecer à CNN e a organização dos mesmos.

Um estimador FR tem como entradas o conteúdo original e o conteúdo degradado e tem como propósito avaliar a qualidade do conteúdo degradado relativamente ao conteúdo original (Figura 5.1). Tendo isto em consideração, os dados de entrada fornecidos têm de dizer respeito ao conteúdo degradado e ao correspondente conteúdo original. Para isso foram gerados dois tipos de imagens compostas.

Um dos tipos de imagens compostas foi construído com as projeções da PC original colocadas nas linhas de cima e as projeções da PC degradada colocadas nas linhas de baixo de uma matriz retangular de quatro por três imagens. A figura 5.4 é um exemplo de uma dessas imagens compostas para a PC *longdress* com uma rotação do cubo de 45° em torno de y .

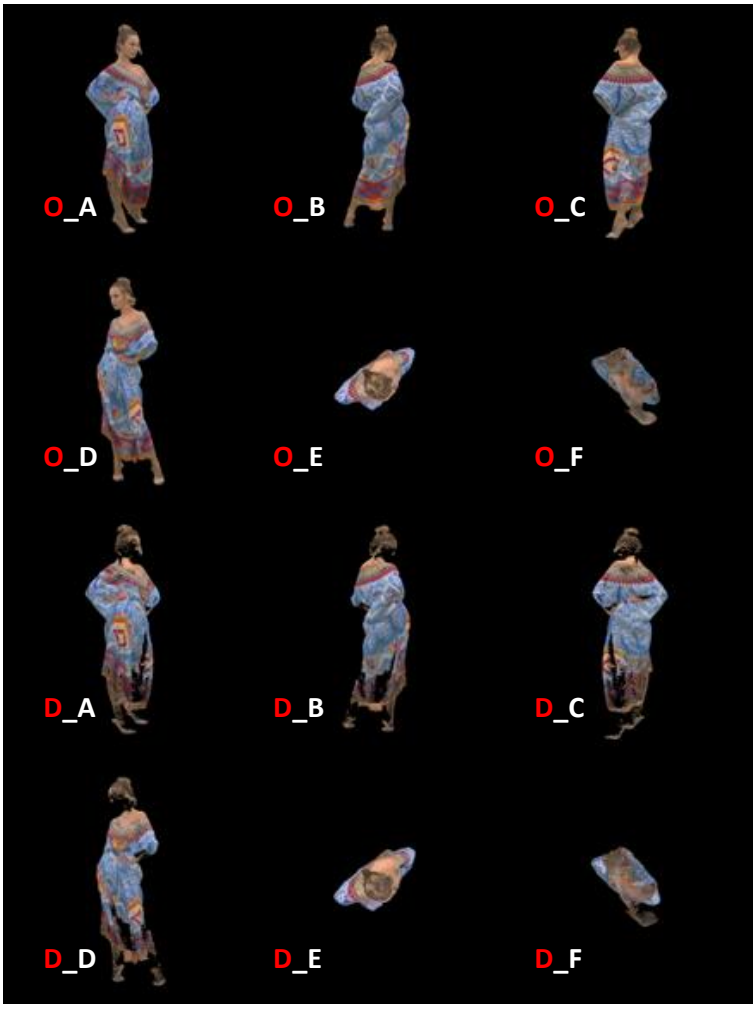


Figura 5.4: Exemplo de imagem composta planar de entrada da CNN.

Nas duas linhas de cima estão dispostas as projeções da PC original, em que a letra "O"a vermelho significa "Original"e a letra à frente representa a que face do cubo a projeção diz respeito. Nas duas linhas de baixo, em que a letra "D"a vermelho significa "Degradada"e a letra à frente representa mais uma vez a face a que corresponde a projeção, estão dispostas as projeções da PC degradada comprimidas com o software *3dtk* com um nível de qualidade baixo. Este tipo de imagem composta tem uma resolução de 583x437 píxeis que resultou na concatenação das imagens das projeções com resolução de 224x224 píxeis, e posteriormente, foi feito um ajuste de tamanho para a imagem composta ocupar apenas 65% da sua resolução original. Isto porque uma CNN tem limitações quanto à resolução das imagens de entrada e para um tamanho superior a este seriam necessários mais recursos computacionais, particularmente memória da *GPU*.

Para treinar a rede pré-treinada *ResNet* em vez de ter sido feito um ajuste de resolução de 65% foi feito um ajuste de maneira a usar apenas 35% da sua resolução original, isto porque a rede *ResNet-101* é a rede mais complexa e mais profunda de todas as redes em teste e apenas funciona para uma resolução máxima de 314x236 antes de esgotar a capacidade de memória da *GPU*.

Como a resolução das imagens de entrada de uma CNN é limitada e quanto menor for menos tempo leva o treino, foi criado outro tipo de imagens compostas semelhante ao anterior mas em vez das imagens serem concatenadas em plano, foram concatenadas em profundidade tal como demonstrado na figura 5.6 em que as letras significam o mesmo que na figura 5.4.

Cada projeção é uma imagem 2D RGB, ou seja é uma matriz de três dimensões de 224x224x3, em que o número três na terceira dimensão diz respeito aos três canais de cor, vermelho, verde e azul, tal como ilustrado na figura 5.5.

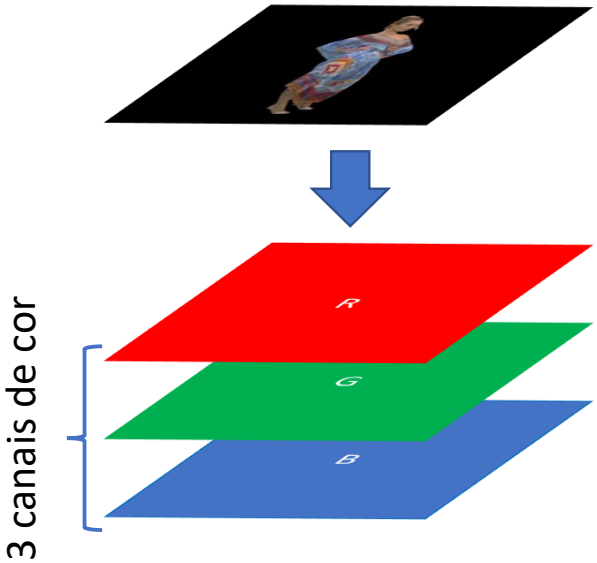


Figura 5.5: Canais de cor de uma projeção.

Esta imagem composta é também uma matriz de 3 dimensões mas com todas as projeções originais e degradadas concatenadas na terceira dimensão da matriz pela ordem representada na figura 5.6. No que resulta uma matriz de 224x224x36, onde 36 são os canais de cor de todas as 12 imagens.

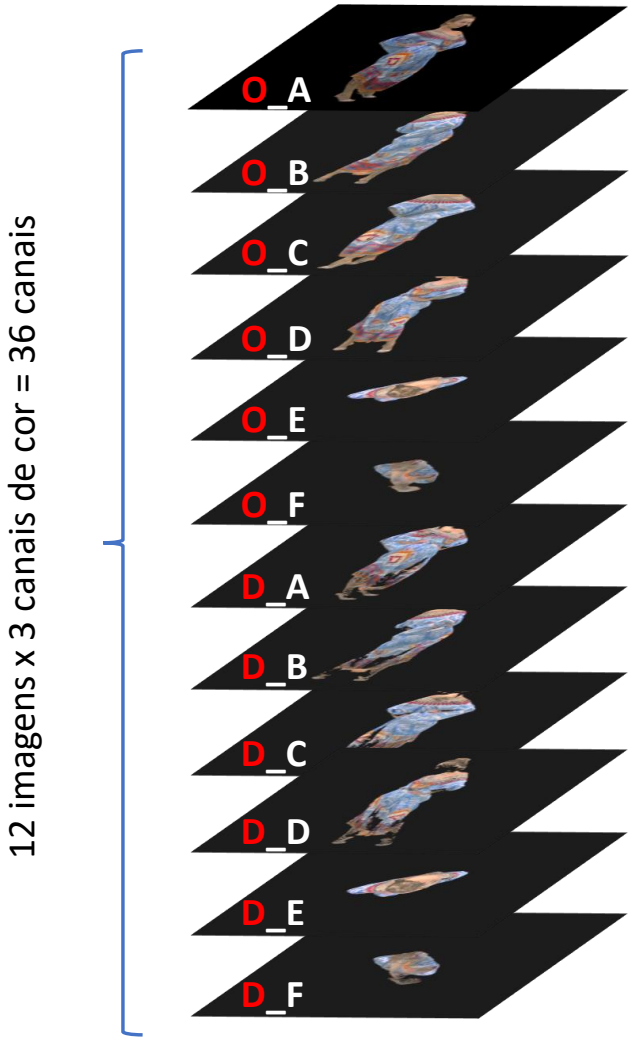


Figura 5.6: Exemplo de imagem composta em profundidade de entrada da CNN.

Estes dois tipos de imagens compostas apresentam duas formas diferentes de organização dos dados de entrada.

Foram gerados estes dois tipos de imagens compostas para todas as PCs de pequena escala *longdress*, *romanoillamp* e *bumbameuboi*, originais e com três níveis de degradação, comprimidas por dois softwares diferentes (*3dtk* e *pcl*), e para cada 1° de rotação em torno de x e y. Isto resultou num total de 12960 imagens compostas por abordagem, como se comprova em 5.1.

Métodos de compressão — Eixos x e y

$$3 \times 2 \times 3 \times 2 \times 360 = 12960 \tag{5.1}$$

Nuvens de pontos — Níveis de degradação — Rotação em torno de x e y

5.2 *Dataset* Suplementar para teste das redes

Como descrito na secção 3.5, foi conduzido um novo estudo de avaliação de qualidade subjetiva de PCs, semelhante ao [17], mas com três novas nuvens,*Loot*, *Statue Klimt* e *Queen*. O objetivo deste estudo é obter mais dados para teste das CNNs permitindo uma verificação adicional do desempenho do estimador, usando dados diferentes dos usados no treino das redes.

Tal como nas PCs da secção anterior foram extraídas as projeções destas novas três PCs, originais e com três níveis de degradação, comprimidas por dois métodos de compressão diferentes (3dtk e pcl). Mas desta vez apenas foram retiradas as seis projeções nas faces do cubo que inclui toda a PC, sem qualquer tipo de rotação. Posteriormente também estas PCs foram organizadas em imagens compostas em plano e imagens compostas em profundidade. Resultando num total de 18 imagens compostas por versão.

5.3 Utilização dos resultados do Teste de Avaliação Subjetiva de Qualidade de Nuvens de Pontos com Textura

Como referido no capítulo 4 este estudo vai incidir apenas em aprendizagem supervisionada. Este tipo de aprendizagem tem como requisito a disponibilidade de dados anotados com as respetivas etiquetas. Neste caso as etiquetas que irão ser associadas às imagens compostas são os resultados MOS obtidos no estudo [17].

No estudo foram recolhidos vários resultados em várias universidades: Universidade da Beira Interior (UBI), Covilhã, Portugal; Universidade de Coimbra, Coimbra, Portugal e University North (UNIN), Varaždin, Croácia. Todos os resultados recolhidos foram utilizados.

Na UBI participaram no estudo 19 observadores, na UC 15 e na UNIN 16. Cada participante avaliou todas as PCs de pequena escala comprimidas pelos dois métodos de compressão e com três níveis de degradação. Foi seguido o método *DSIS simultaneous* com uma escala de avaliação de 5 níveis.

Todos os participantes avaliaram os conteúdos *bumbameuboi*, *longdress*, *romanoillamp*, *ucl*, *citiusp*, *ipanemacut* e *ramos*, referidos na secção 3.4. Apenas interessam os resultados que dizem respeito às nuvens de pequena escala. Esses resultados foram selecionados e encontram-se listados nas tabelas 5.1, 5.2 e 5.3. Relembrando-se que a escala usada vai de 1 a 5 conforme a tabela 3.2

Tabela 5.1: Universidade da Beira Interior (UBI), Covilhã, Portugal

Conteúdo	**Tipo de compressão**	**Qualidade da compressão**	**Participante**																		
			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
bumbameuboi	**original**	–	5	5	5	5	4	5	5	5	5	5	4	5	5	5	3	5	5	5	5
	3dtk	**Elevada**	3	3	4	3	3	4	3	4	2	1	2	2	3	2	1	3	3	2	3
		Média	2	2	2	2	2	2	1	2	1	1	2	2	2	1	1	1	1	2	1
		Baixa	1	1	1	2	1	1	2	2	1	1	2	1	3	1	1	1	1	1	1
	pcl	**Elevada**	4	4	4	5	4	5	5	5	5	5	4	5	4	4	4	4	5	3	5
		Média	3	3	3	5	3	2	4	3	4	3	3	5	4	4	3	4	4	3	4
		Baixa	2	1	1	3	1	1	3	3	1	2	2	3	4	2	1	1	2	1	1
longdress	**original**	–	5	5	5	5	5	5	4	5	5	5	5	5	5	4	5	5	5	5	5
	3dtk	**Elevada**	5	5	5	5	5	5	5	5	5	5	5	5	3	5	4	5	5	5	5
		Média	2	1	2	3	2	1	1	2	1	1	2	1	2	1	1	1	2	1	1
		Baixa	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	pcl	**Elevada**	5	5	5	5	5	5	5	5	5	5	4	5	4	5	3	5	5	5	4
		Média	4	5	5	5	5	5	5	5	5	5	5	5	4	5	3	5	5	5	5
		Baixa	4	4	3	4	4	4	4	2	4	3	4	5	4	5	2	4	5	3	4
romanoillamp	**original**	–	5	5	5	5	5	5	5	5	5	5	5	5	4	5	4	5	4	5	4
	3dtk	**Elevada**	3	5	4	4	5	4	4	4	4	4	4	4	3	4	4	4	5	4	5
		Média	2	2	2	3	2	3	3	2	2	1	3	1	2	1	1	3	1	2	2
		Baixa	1	2	2	3	1	3	1	2	1	1	2	1	2	1	1	1	1	1	1
	pcl	**Elevada**	5	5	5	5	5	5	4	5	5	5	4	5	4	4	2	5	5	5	5
		Média	4	5	5	5	4	5	5	5	5	4	5	5	3	5	5	5	5	4	5
		Baixa	4	4	4	4	4	4	5	4	4	3	4	5	5	4	2	5	4	4	4

Tabela 5.2: Resultados Universidade de Coimbra (UC), Coimbra, Portugal

Conteúdo	Tipo de compressão	Qualidade da compressão	Participante														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
bumbameuboi	original	–	5	4	5	5	5	3	5	5	5	5	5	5	5	5	5
	3dtk	Elevada	1	2	3	1	3	1	4	1	2	2	3	3	3	3	2
		Média	1	2	1	1	2	2	3	1	1	2	2	2	2	3	2
		Baixa	1	1	1	1	1	3	3	1	1	1	1	2	3	1	1
	pcl	Elevada	5	4	5	5	5	5	5	4	3	5	5	5	5	5	5
		Média	5	3	2	5	3	4	5	2	3	2	3	3	3	4	3
		Baixa	1	3	1	1	2	1	4	1	1	3	1	3	1	3	2
longdress	original	–	5	5	5	5	5	4	5	5	4	5	4	5	5	5	5
	3dtk	Elevada	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
		Média	1	1	1	1	2	1	3	1	1	2	1	1	1	1	2
		Baixa	1	1	1	1	1	1	3	1	1	1	1	1	1	1	1
	pcl	Elevada	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5
		Média	5	4	5	5	5	4	5	5	3	5	5	5	5	5	5
		Baixa	4	3	4	5	4	4	5	5	5	4	2	3	2	4	4
romanoillamp	original	–	5	4	5	5	5	4	5	5	2	5	5	5	5	5	5
	3dtk	Elevada	5	3	5	5	4	4	5	5	5	4	5	4	5	5	4
		Média	4	2	3	1	3	1	4	1	4	4	2	2	1	3	2
		Baixa	3	2	1	5	1	2	3	1	1	3	1	2	1	3	3
	pcl	Elevada	5	5	5	5	5	5	5	5	3	5	5	5	5	5	5
		Média	5	5	5	5	5	5	5	5	3	5	5	5	5	5	5
		Baixa	4	4	5	5	4	4	5	5	4	5	4	4	5	5	5

Tabela 5.3: Resultados University North (UNIN), Varaždin, Croácia

Conteúdo	Tipo de compressão	Qualidade da compressão	Participante															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
bumbameuboi	original	–	5	5	5	5	4	5	4	5	5	5	4	4	3	4	5	4
	3dtk	Elevada	2	3	3	3	1	1	2	2	3	3	2	2	2	2	2	2
		Média	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	2
		Baixa	2	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1
	pcl	Elevada	5	5	5	5	4	5	4	5	4	5	4	4	5	3	5	3
		Média	4	3	5	5	4	3	3	3	2	2	2	4	5	1	2	2
		Baixa	2	1	1	2	2	2	2	1	1	1	1	1	2	1	1	2
longdress	original	–	5	5	5	5	4	5	5	5	5	5	5	5	5	5	5	5
	3dtk	Elevada	5	4	5	5	5	5	5	5	5	5	5	4	5	5	5	5
		Média	1	2	1	2	2	1	1	2	1	1	1	1	1	1	1	1
		Baixa	1	2	1	2	1	1	1	1	1	1	1	1	1	1	1	1
	pcl	Elevada	5	4	5	5	5	5	5	5	5	5	4	5	5	5	5	5
		Média	5	4	5	5	5	5	5	4	4	5	2	4	3	4	5	5
		Baixa	2	3	4	4	4	3	4	4	3	3	3	4	3	1	4	2
romanoillamp	original	–	5	5	5	5	5	5	5	5	5	5	4	5	5	5	5	5
	3dtk	Elevada	3	3	4	4	3	4	5	2	3	3	3	5	3	4	4	3
		Média	2	2	2	2	2	1	2	2	2	3	2	2	1	2	1	2
		Baixa	1	2	1	2	3	1	1	1	1	1	2	2	1	2	1	1
	pcl	Elevada	5	5	5	5	5	5	5	5	5	5	4	3	5	5	5	5
		Média	5	4	5	5	5	4	4	4	5	5	4	4	5	4	5	5
		Baixa	5	3	5	5	4	5	3	4	4	3	2	3	4	1	4	3

Os resultados que dizem respeito às sequências originais não vão ser tidos em consideração uma vez que estamos interessados em medir a degradação dos conteúdos. Para os outros resultados foram calculados os seus valores médios para cada PC (MOS) sendo estes os valores que foram utilizados como etiqueta. Estes valores estão indicados na tabela 5.4.

Tabela 5.4: Resultados MOS

Conteúdo	Tipo de compressão	Qualidade da compressão	MOS
bumbameuboi	***3dtk***	**Elevada**	2.4
		Média	1.6
		Baixa	1.3
	pcl	**Elevada**	4.52
		Média	3.34
		Baixa	1.72
longdress	***3dtk***	**Elevada**	4.9
		Média	1.36
		Baixa	1.14
	pcl	**Elevada**	4.84
		Média	4.64
		Baixa	3.62
romanoillamp	***3dtk***	**Elevada**	4.04
		Média	2.1
		Baixa	1.66
	pcl	**Elevada**	4.76
		Média	4.7
		Baixa	4.06

5.4 Utilização dos resultados do Teste Suplementar de Avaliação Subjetiva de Qualidade de Nuvens de Pontos com Textura

Com o intuito de confirmar os resultados obtidos pelas redes treinadas foi realizado mais um estudo para ser usado apenas como teste. Este estudo foi realizado na Universidade de Coimbra, Coimbra, Portugal e contou com um total de 20 participantes. Cada participante avaliou todas as três PCs, *Loot*, *Statue Klimt* e *Queen*, comprimidas pelos dois softwares e com três níveis de degradação. Foi seguido o método *DSIS simultaneous* com uma escala de avaliação de 5 níveis, de 1 a 5, conforme a tabela 3.2. Os resultados estão listados na tabela 5.5.

Tal como no caso anterior os resultados que dizem respeito às sequências originais não vão ser tidos em consideração. Para os outros foram calculados os seus valores MOS e os intervalos de confiança de 95% assumindo uma distribuição T-Student's. Estes resultados encontram-se na seguinte tabela 5.6

Tabela 5.5: Resultados Universidade de Coimbra (UC), Coimbra, Portugal

Conteúdo	**Tipo de compressão**	**Qualidade da compressão**	**Participante**																			
			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
loot	**original**	–	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	3	5
	3dtk	**Elevada**	5	4	4	5	5	5	4	5	5	5	5	5	5	5	4	5	5	5	4	5
		Média	2	1	2	3	2	3	1	2	1	2	2	3	2	2	2	2	1	1	1	1
		Baixa	2	1	1	3	1	1	1	2	1	1	2	3	1	1	1	1	1	1	1	1
	pcl	**Elevada**	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	4	5	4	5
		Média	4	5	5	5	4	5	5	4	4	4	5	5	5	5	4	4	5	5	3	5
		Baixa	3	3	2	4	3	4	4	5	3	3	4	5	4	4	3	3	4	3	3	4
queen	**original**	–	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5
	3dtk	**Elevada**	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5
		Média	2	1	1	3	1	2	1	2	1	2	2	3	2	2	1	2	2	1	1	1
		Baixa	1	1	1	2	1	1	1	2	1	2	2	2	1	1	2	1	1	1	1	1
	pcl	**Elevada**	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5	5	4	4
		Média	5	4	4	5	4	5	5	5	4	4	5	4	5	5	5	5	4	4	4	3
		Baixa	3	3	3	4	2	4	3	5	2	4	4	4	3	3	3	3	4	3	5	4
statue_ klimt	**original**	–	5	4	3	5	4	4	5	1	5	5	5	5	5	4	2	5	4	5	4	5
	3dtk	**Elevada**	5	3	4	5	5	3	5	4	5	4	5	5	4	3	3	5	5	5	4	5
		Média	3	2	3	3	1	3	2	2	3	3	3	5	3	1	2	2	3	2	4	2
		Baixa	2	1	1	2	1	2	1	2	1	1	3	1	1	1	1	1	1	1	1	1
	pcl	**Elevada**	3	3	5	5	4	4	5	5	5	5	5	5	5	4	3	5	4	5	4	5
		Média	5	3	4	5	5	4	4	5	5	4	5	5	5	4	3	5	5	5	4	5
		Baixa	4	4	1	5	3	5	4	4	3	3	3	5	5	3	2	4	3	5	2	4

Tabela 5.6: Resultados MOS e intervalos de confiança

Conteúdo	Tipo de compressão	Qualidade da compressão	MOS	CI
loot	***3dtk***	**Elevada**	4.75	0.207921
		Média	1.8	0.325669
		Baixa	1.35	0.313954
	pcl	**Elevada**	4.85	0.171456
		Média	4.55	0.283058
		Baixa	3.55	0.355295
queen	***3dtk***	**Elevada**	4.9	0.144052
		Média	1.65	0.313954
		Baixa	1.3	0.220043
	pcl	**Elevada**	4.85	0.171456
		Média	4.45	0.283058
		Baixa	3.45	0.386382
statue_klimt	***3dtk***	**Elevada**	4.35	0.380368
		Média	2.6	0.440086
		Baixa	1.3	0.267349
	pcl	**Elevada**	4.45	0.355295
		Média	4.5	0.32211
		Baixa	3.6	0.534698

Estes resultados foram usados para cálculo dos fatores de desempenho finais de teste, de maneira a poder aferir o desempenho das CNNs treinadas com dados nunca antes vistos.

5.5 Atribuição de etiquetas

Foram testados dois tipos de dados de entrada da rede, as imagens compostas em plano e as imagens compostas em profundidade.

Para cada imagem composta individual foi-lhe associada a etiqueta do seu valor MOS correspondente.

Por exemplo para a imagem composta da figura 5.4 que representa a PC *longdress* comprimida com o software *3dtk* com um nível de qualidade baixo, foi-lhe associada a etiqueta de 1.14, assim como a todos as imagens compostas que dizem respeito a esta PC comprimida com este software e este nível de qualidade. Este valor de etiqueta foi então atribuído a 720 imagens compostas, que dizem respeito a este mesmo conteúdo, representantes das rotações de 1° em 1° em torno de x e de y referidas na secção 5.1.1.

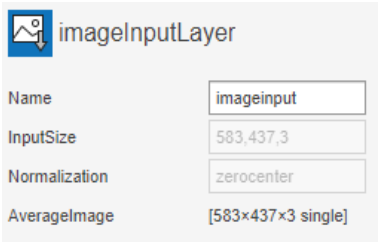
5.6 Modificações nas arquiteturas das CNNs

Tal como referido na secção 4.4 este estudo vai testar três redes pré-treinadas usando *transfer learning* (*AlexNet*, *GoogLeNet* e *ResNet*). Para isto todas elas tiveram de sofrer alterações na sua arquitetura, uma vez que os dados de entrada deste problema são diferentes de imagens 2D comuns e também porque estas redes foram contruídas com o intuito de resolver problemas de classificação e não de regressão.

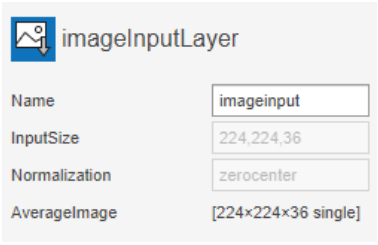
O software utilizado para treino e teste das CNNs foi o "*MATLAB R2019a*"com as *toolboxs "Deep Learning Toolbox"e "Parallel Computing Toolbox"* instaladas. Todas as redes analisadas estão disponíveis no *MATLAB* em forma de *add-on* sendo necessária a sua prévia instalação.

Em todas as redes testadas as alterações na sua arquitetura foram sensivelmente as mesmas.

Começando pela primeira camada, a camada de entrada de imagens tem um parâmetro configurável, a resolução das imagens de entrada, como mostra a figura 5.7.



(a) Redes com imagens compostas em plano.

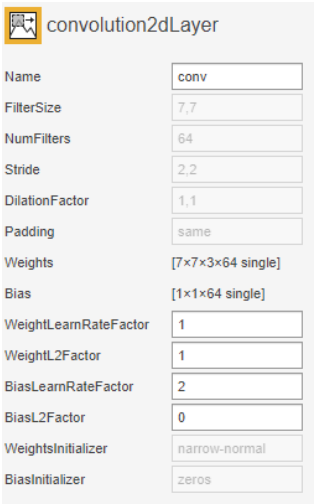


(b) Redes com imagens compostas em profundidade.

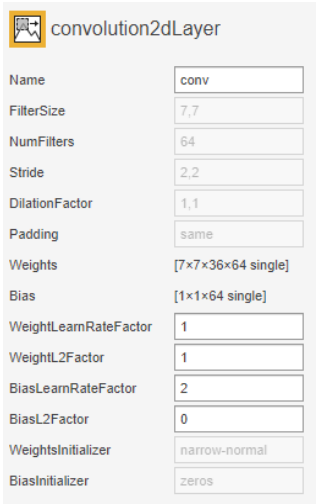
Figura 5.7: Camadas de entrada.

Para as imagens compostas em plano a resolução tem de ser alterada para 583,437,3, sendo que o 3 diz respeito a cada canal de cor RGB 5.7a. Para as imagens compostas em profundidade a resolução tem de ser alterada para 224,224,36, em que o 36 corresponde aos três canais de cor de cada uma das 12 imagens concatenadas na terceira dimensão da matriz 5.7b.

De seguida foi necessário modificar o tamanho dos filtros da primeira camada convolucional das redes que vão processar as imagens compostas em profundidade. Para uma rede em que a entrada são imagens RGB, como as imagens compostas em plano, não é preciso fazer modificações uma vez que os filtros da primeira camada convolucional já contemplam os três canais de cor presentes neste tipo de imagens, exemplificado na figura 5.8a. Para as imagens compostas em profundidade é preciso modificar o tamanho dos filtros para [7x7x36x64] para que o filtro tenha em consideração todos os 36 canais de cor das imagens compostas em profundidade. A figura 5.8b é exemplo desta modificação, e diz respeito à primeira camada de convolução da GoogleNet.



(a) Redes com imagens compostas em plano.



(b) Redes com imagens compostas em profundidade.

Figura 5.8: Primeira camada convolução (exemplo retirado de GoogleNet).

Por fim, foi necessário alterar as camadas finais das redes testadas, para isso foi desenvolvido um novo conjunto de camadas finais, constituído por uma camada *fully connected* com saída de 1000 unidades seguida

de uma camada de retificação linear, e uma camada de *dropout* de 50%. A seguir a estas vem uma nova camada *fully connected* com saída de 100 unidades, mais uma vez seguida de uma *ReLU* e uma de *dropout* de 50%. Por fim foram implementadas duas camadas *fully connected* consecutivas. A primeira com saída de 10 unidades e a última com saída de uma unidade, uma vez a camada de regressão só pode ter uma entrada e uma saída. Esta é seguida de uma nova camada de retificação linear e finalmente entra na camada de regressão com a função de perda de erro médio quadrático. Estas camadas finais serviram para adaptar esta rede ao problema de regressão. Foram usadas camadas *fully connected* com função ativação *ReLU*, seguidas de camadas de *dropout* para prevenção de *overfitting* e para aceleração de cálculos. Estas camadas finais estão representadas na figura 5.9.

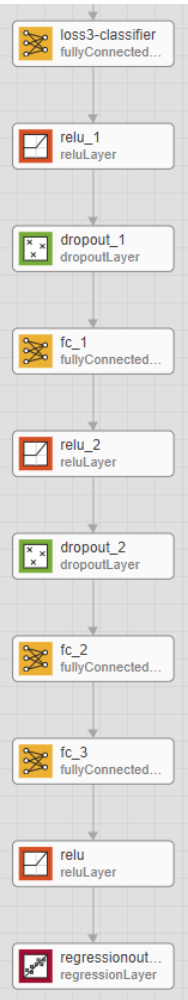


Figura 5.9: Camadas finais da CNN (*MATLAB*).

Todas estas modificações referidas nesta secção foram implementadas de forma idêntica nas redes testadas.

5.7 *AlexNet* no *MATLAB*

Pelo facto da *AlexNet* ser a rede mais simples a implementar foi a primeira a ser testada. Para isto foi necessário instalar o *add-on* respetivo a esta rede *"Deep Learning Toolbox Model for AlexNet Network"*. A rede implementada tem a arquitetura que foi analisada na secção 4.4.1.

Foram feitas as adaptações necessárias referidas na secção anterior o que resultou numa rede com um

total de 32 camadas. Cinco camadas de convolução, duas camadas de normalização de resposta local a seguir às duas primeiras camadas de convolução. Três camadas *pooling* máximo a seguir a cada grupo de convoluções. Seis camadas *fully connected* nas camadas finais. Dez camadas ReLU pois todas as camadas de convolução e *fully connected* tem como função ativação *ReLU*. Por fim quatro camadas de *dropout* nas camadas finais para prevenção de *overfitting* e para aceleração de cálculos. Todas estas camadas resultaram numa rede com uma profundidade de 11.

5.8 *GoogleNet* no *MATLAB*

Uma das redes com melhores resultados em problemas de classificação de imagens é a *GoogleNet*, razão pela qual foi uma das redes mais explorada neste estudo.

Mais uma vez foi necessário instalar o *add-on* respetivo a esta rede *"Deep Learning Toolbox Model for GoogLeNet Network"*. Esta rede tem a arquitetura que foi analisada de forma detalhada na secção 4.4.2 e foi adaptada consoante a descrição da secção 5.6. No que resultou uma rede com 151 camadas com nove *inception modules* com uma profundidade de 2 cada um. Estas adaptações resultaram numa rede com uma profundidade total de 25.

5.9 *ResNet* no *MATLAB*

Outra rede que apresentou bons resultados em classificação de imagens foi a *ResNet* e foi também uma das redes mais testadas. Para isto foi necessário instalar o *add-on "Deep Learning Toolbox Model for ResNet-101 Network"*. A arquitetura desta rede foi descrita na secção 4.4.3 e foi adaptada nas camadas inicial e finais de acordo com a descrição da secção 5.6 mas como referido na secção 5.1.2 para as imagens compostas em plano a resolução não vai ser 583x437 mas sim de 314x236.

Isto resultou numa rede de 354 camadas e com profundidade total de 104, sendo a rede mais profunda presente neste estudo.

6 Resultados Experimentais

Todas as experiências foram realizadas numa máquina com um processador *Intel Core i5-7600* (3.50 GHz) com uma capacidade de memória *RAM* de 32 GB com o sistema operativo *Windows 10 Pro* de 64 bits. Esta máquina contém uma placa gráfica *MSI GeForce RTX 2070 GAMING 8G* com 8 GB de memória dedicada e com um *boost clock* de 1620 MHz. Todo o processamento das redes foi efetuado recorrendo ao poder computacional da *GPU* da placa gráfica.

Todas as redes com as adaptações descritas no capítulo 5 foram testadas com as imagens compostas em plano e em profundidade, com os mesmos hiper-parâmetros e com o *dataset* dividido da mesma forma, com intuito de garantir que os resultados são comparáveis entre si.

Para confirmar a validade e qualidade dos resultados obtidos foram calculados os seguintes fatores de desempenho em comparação com os resultados subjetivos considerando estes como referência: coeficiente de correlação de Pearson (PCC), coeficiente de correlação de Spearman (SROCC) e a raíz do erro médio quadrático (RMSE). Estes fatores foram calculados tanto para as previsões feitas pela rede para o *dataset* de validação como para o de teste. Para além disto foi testado um *dataset* suplementar de teste com dados nunca vistos pela rede, descrito na secção 5.2. Para este *dataset* foram calculados os seguintes parâmetros de desempenho: coeficiente de correlação de Pearson (PCC), coeficiente de correlação de Spearman (SROCC), a raíz do erro médio quadrático (RMSE) e rácio de *outliers* (OR). Este teste extra serviu para observar o comportamento da rede com nuvens diferentes das do primeiro *dataset*.

6.1 Treino, Validação e Teste

Tanto para as imagens compostas em plano como para as em profundidade o primeiro *dataset* resulta num total de 12960 imagens compostas para cada um dos tipos de imagens. Este *dataset* foi dividido da seguinte forma, 80% dos 12960 foram dedicados a treino e validação da rede e 20% foram reservados para teste da rede. Por sua vez 80% dos 80% dedicados a treino e validação foram para treino e 20% para validação. Esta divisão está esquematizada na figura 6.1.

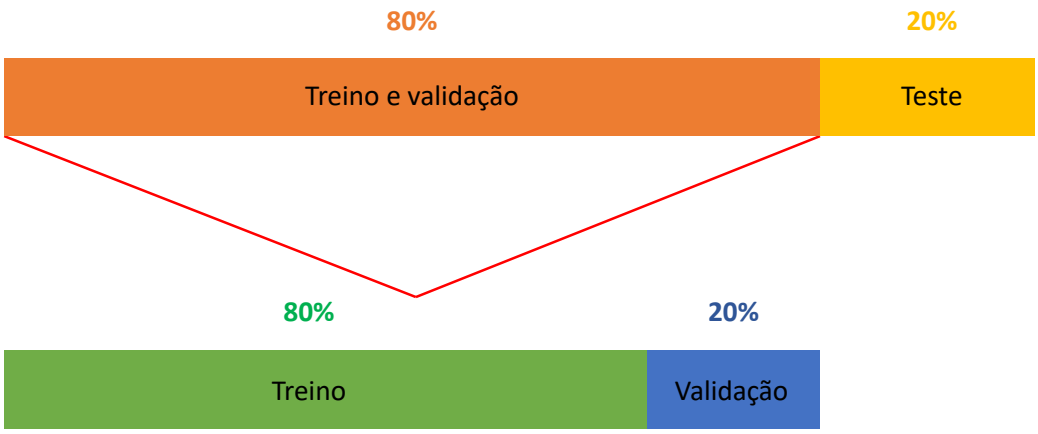


Figura 6.1: Divisão do *dataset*.

Quanto aos hiper-parâmetros definidos para estes testes encontram-se descritos na seguinte tabela 6.1.

Tabela 6.1: Hiper-parâmetros.

Parâmetro	**Valor**
Tamanho do *Minibatch*	32 / 16 / 8
Taxa de aprendizagem	0.0001
Épocas	30
Iterações por época	258 / 517 / 1035
Frequência de validação	1x por época

Para as três redes e para as duas versões de imagens compostas foram feitos os testes para cada valor de tamanho *minibatch* descrito na tabela para uma taxa de aprendizagem de 0,0001 durante trinta épocas.

Na Figura 6.2 está representado em forma de fluxograma o algoritmo utilizado para treino das CNNs. O fluxograma tem início no retângulo "Primeiro MB da época (Dataset de treino)", sendo que MB significa *minibatch*, e fim no retângulo "Fim do treino".

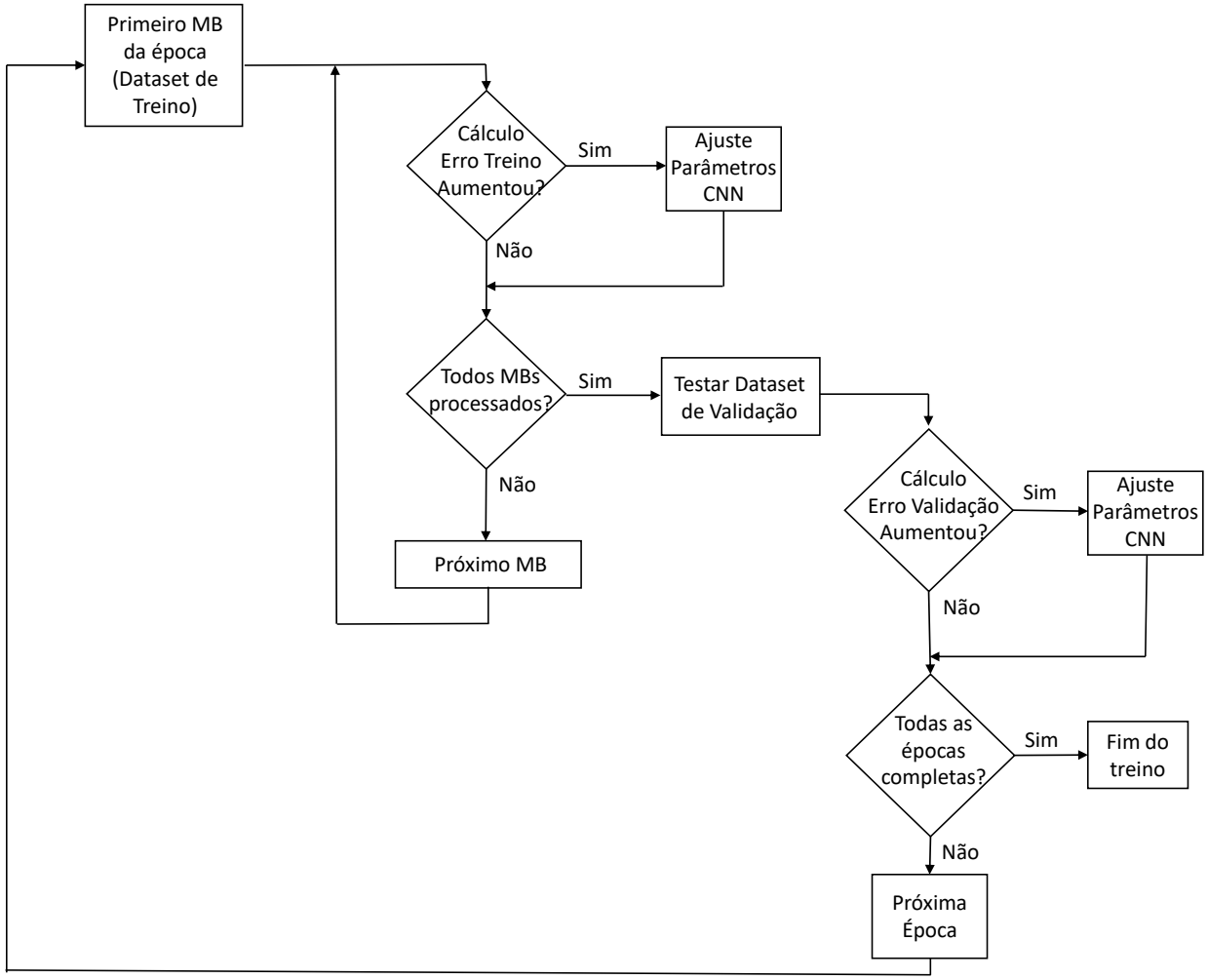


Figura 6.2: Fluxograma do algoritmo de treino.

O tempo de treino para cada uma das redes testadas está listado na tabela 6.2.

Tabela 6.2: Tempo de treino das CNNs.

Redes	***Tempo de treino (Horas:Minutos:Segundos)***
AlexNet / Minibatch 32 / Imagens compostas em plano	01:00:32
AlexNet / Minibatch 16 / Imagens compostas em plano	00:58:22
AlexNet / Minibatch 8 / Imagens compostas em plano	01:39:11
AlexNet / Minibatch 32 / Imagens compostas em profundidade	00:56:45
AlexNet / Minibatch 16 / Imagens compostas em profundidade	01:15:08
AlexNet / Minibatch 8 / Imagens compostas em profundidade	01:22:00
GoogLeNet / Minibatch 32 / Imagens compostas em plano	09:56:42
GoogLeNet / Minibatch 16 / Imagens compostas em plano	03:44:47
GoogLeNet / Minibatch 8 / Imagens compostas em plano	08:30:37
GoogLeNet / Minibatch 32 / Imagens compostas em profundidade	01:25:25
GoogLeNet / Minibatch 16 / Imagens compostas em profundidade	02:08:12
GoogLeNet / Minibatch 8 / Imagens compostas em profundidade	03:29:47
ResNet-101 / Minibatch 32 / Imagens compostas em plano	04:52:37
ResNet-101 / Minibatch 16 / Imagens compostas em plano	06:53:24
ResNet-101 / Minibatch 8 / Imagens compostas em plano	11:29:56
ResNet-101 / Minibatch 32 / Imagens compostas em profundidade	04:25:09
ResNet-101 / Minibatch 16 / Imagens compostas em profundidade	05:52:56
ResNet-101 / Minibatch 8 / Imagens compostas em profundidade	10:45:35
Total	80:47:05

O tempo total de treino de todas estas redes foi de 3 dias 8 horas 47 minutos e 5 segundos. A rede que demorou mais tempo a treinar foi a *ResNet-101 / Minibatch 8 / Imagens compostas em plano* com um tempo total de 11 horas 29 minutos e 56 segundos, por oposição a rede que demorou menos tempo a treinar foi a *AlexNet / Minibatch 32 / Imagens compostas em profundidade* com um tempo total de 56 minutos e 45 segundos.

Após a realização do treino, validação e teste no primeiro *dataset*, foi feito um teste final com o *dataset* suplementar e calculados os fatores de desempenho.

6.2 Resultados do *Dataset* Principal para a *AlexNet*

A rede *AlexNet* com a arquitetura definida na secção 5.7 foi treinada para as duas versões de imagens compostas usando o *dataset* descrito na secção 5.1.2, dividido da forma representada na secção 6.1 e com os hiper-parâmetros definidos na mesma secção.

6.2.1 Imagens compostas em plano

Na tabela 6.3 estão representados todos os fatores de desempenho para um *minibatch* de 32, um *minibatch* de 16 e um *minibatch* de 8. A **negrito** estão representados os melhores resultados obtidos comparados entre tamanhos de *minibatches*.

Tabela 6.3: Fatores de desempenho para a *AlexNet* com imagens compostas em plano.

Dataset	***Fator de desempenho***	***Tamanho do minibatch***		
		32	***16***	***8***
Validação	***RMSE***	**0.63985**	0.7379	0.81155
	PCC	**0.9399**	0.898	0.8675
	SROCC	**0.9338**	0.9297	0.9141
Teste	***RMSE***	1.0495	**0.8892**	0.8966
	PCC	0.795	**0.8534**	0.8316
	SROCC	0.8435	**0.9157**	0.91

Para o *dataset* de validação os melhores resultados foram obtidos com um tamanho de *minibatch* de 32. Para o *dataset* de teste o tamanho de *minibatch* que obteve melhores resultados foi o de 16.

6.2.2 Imagens compostas em profundidade

Na tabela 6.4 estão representados todos os fatores de desempenho para um *minibatch* de 32, um *minibatch* de 16 e um *minibatch* de 8. A **negrito** estão representados os melhores resultados obtidos comparados entre tamanhos de *minibatches*.

Tabela 6.4: Fatores de desempenho para a *AlexNet* com imagens compostas em profundidade.

Dataset	***Fator de desempenho***	***Tamanho de minibatch***		
		32	***16***	***8***
Validação	***RMSE***	**0.97485**	1.1435	1.0934
	PCC	**0.7923**	0.7051	0.7285
	SROCC	**0.7374**	0.5483	0.5162
Teste	***RMSE***	**0.5559**	0.7697	1.1095
	PCC	**0.9703**	0.9376	0.8165
	SROCC	**0.974**	0.9394	0.9472

Tal como nas imagens compostas em plano os melhores resultados para o *dataset* de validação são para um tamanho de *minibatch* de 32 e os melhores resultados obtidos no *dataset* de teste são também para um tamanho de *minibatch* de 32.

Comparando os resultados de imagens compostas em plano com os resultados de imagens compostas em profundidade, os resultados são melhores com imagens compostas em plano para o *dataset* de validação mas são melhores com imagens compostas em profundidade para o *dataset* de teste.

6.3 Resultados do *Dataset* Principal para a *GoogLeNet*

A rede *GoogLeNet* com a arquitetura definida na secção 5.8 foi treinada para as duas versões de imagens compostas usando o *dataset* descrito na secção 5.1.2, dividido da forma representada na secção 6.1 e com os hiper-parâmetros definidos na mesma secção.

6.3.1 Imagens compostas em plano

Na tabela 6.5 estão representados todos os fatores de desempenho para um *minibatch* de 32, um *minibatch* de 16 e um *minibatch* de 8. A **negrito** estão representados os melhores resultados obtidos comparados entre tamanhos de *minibatches*.

Tabela 6.5: Fatores de desempenho para a *GoogLeNet* com imagens compostas em plano.

Dataset	***Fator de desempenho***	***Tamanho do minibatch***		
		32	***16***	***8***
Validação	***RMSE***	0.55913	0.57526	**0.52854**
	PCC	0.9209	0.9163	**0.9323**
	SROCC	**0.9417**	0.9379	0.9326
Teste	***RMSE***	1.0326	1.0369	**0.9231**
	PCC	0.778	0.792	**0.8436**
	SROCC	0.7953	0.8708	**0.9068**

Para a *GoogLeNet* com imagens compostas em plano os melhores resultados para o *dataset* de validação foram para um tamanho de *minibatch* de 8 com exceção do coeficiente de correlação de Spearman que foi

melhor para um tamanho de *minibatch* de 32. Os melhores resultados para o *dataset* de teste foram obtidos com um tamanho de *minibatch* de 8.

6.3.2 Imagens compostas em profundidade

Na tabela 6.6 estão representados todos os fatores de desempenho para um *minibatch* de 32. um *minibatch* de 16 e um *minibatch* de 8. A **negrito** estão representados os melhores resultados obtidos comparados entre tamanhos de *minibatches*.

Tabela 6.6: Fatores de desempenho para a *GoogLeNet* com imagens compostas em profundidade.

Dataset	***Fator de desempenho***	***Tamanho do minibatch***		
		32	***16***	***8***
Validação	***RMSE***	0.66874	0.77856	**0.55557**
	PCC	0.8963	0.8628	**0.9292**
	SROCC	**0.9045**	0.8796	0.8923
Teste	***RMSE***	**0.2067**	0.2293	0.2213
	PCC	**0.9961**	0.9925	0.9911
	SROCC	**0.9587**	0.946	0.9442

Os melhores resultados para as imagens compostas em profundidade da *GoogLeNet* para o *dataset* de validação são com tamanho de *minibatch* de 8 com exceção do coeficiente de correlação de Spearman que é melhor para um tamanho de *minibatch* de 32. Para o *dataset* de teste os resultados são melhores para um tamanho de *minibatch* de 32.

Comparando os resultados de imagens compostas em plano com os resultados de imagens compostas em profundidade, e tal como se verificou na *AlexNet*, as imagens compostas em plano apresentam melhores resultados para o *dataset* de validação e as imagens compostas em profundidade apresentam melhores resultados para o *dataset* de teste.

6.4 Resultados do *Dataset* Principal para a *ResNet-101*

A rede *ResNet-101* com a arquitetura definida na secção 5.9 foi treinada para as duas versões de imagens compostas, sendo que as imagens compostas em plano para esta rede têm uma resolução de 314x236, usando o *dataset* descrito na secção 5.1.2, dividido da forma representada na secção 6.1 e com os hiper-parâmetros definidos na mesma secção.

6.4.1 Imagens compostas em plano

Na tabela 6.7 estão representados todos os fatores de desempenho para um *minibatch* de 32, um *minibatch* de 16 e um *minibatch* de 8. A **negrito** estão representados os melhores resultados obtidos comparados entre tamanhos de *minibatches*.

Tabela 6.7: Fatores de desempenho para a *ResNet-101* com imagens compostas em plano.

Dataset	***Fator de desempenho***	***Tamanho de minibatch***		
		32	***16***	***8***
Validação	***RMSE***	0.73647	0.6317	**0.61465**
	PCC	0.9187	**0.9194**	0.914
	SROCC	0.8953	0.8911	**0.9396**
Teste	***RMSE***	1.2159	1.1558	**1.0607**
	PCC	0.7702	0.7744	**0.804**
	SROCC	0.8471	0.8624	**0.8787**

Para a *ResNet-101* com imagens compostas em plano os melhores resultados para o *dataset* de validação foram para um tamanho de *minibatch* de 8 com exceção do coeficiente de correlação de Pearson que foi melhor para um tamanho de *minibatch* de 16. Os melhores resultados para o *dataset* de teste foram obtidos com um tamanho de *minibatch* de 8.

6.4.2 Imagens compostas em profundidade

Na tabela 6.8 estão representados todos os fatores de desempenho para um *minibatch* de 32, um *minibatch* de 16 e um *minibatch* de 8. A **negrito** estão representados os melhores resultados obtidos comparados entre tamanhos de *minibatches*

Tabela 6.8: Fatores de desempenho para a *ResNet-101* com imagens compostas em profundidade.

Dataset	***Fator de desempenho***	***Tamanho de minibatch***		
		32	***16***	***8***
Validação	***RMSE***	0.92916	1.1362	**0.90493**
	PCC	0.8186	0.736	**0.8335**
	SROCC	0.8471	0.6214	**0.8734**
Teste	***RMSE***	**0.6026**	0.8094	0.7415
	PCC	**0.9632**	0.9252	0.9317
	SROCC	**0.9619**	0.9492	0.953

Os melhores resultados para as imagens compostas em profundidade da *ResNet-101* para o *dataset* de validação são com tamanho de *minibatch* de 8. Para o *dataset* de teste os resultados são melhores para um tamanho de *minibatch* de 32.

Comparando os resultados de imagens compostas em plano com os resultados de imagens compostas em profundidade, e tal como se verificou nas outras redes, as imagens compostas em plano apresentam melhores resultados para o *dataset* de validação e as imagens compostas em profundidade apresentam melhores resultados para o *dataset* de teste.

6.5 Melhores Resultados do *Dataset* Principal

Foi feita uma seleção das redes que apresentam melhores resultados dentro de cada tipo de rede para o *dataset* de validação e para o *dataset* de teste. Estas redes estão presentes na tabela 6.9. Em que as três primeiras linhas são as redes que apresentam melhores resultados para o *dataset* de validação, e as três últimas linhas são as redes que apresentam melhores resultados para o *dataset* de teste. A **negrito** estão representados os melhores resultados para cada fator de desempenho, para o *dataset* de teste e para o de validação.

Dando mais peso aos resultados com o *dataset* de teste é possível concluir que a rede que apresenta melhores resultados é a *GoogLeNet* que utiliza as imagens compostas em profundidade e com um tamanho de *minibatch* de 32. Também é possível verificar que os resultados com o *dataset* de validação dessa rede não ficam muito atrás dos melhores resultados obtidos com as outras redes para esse *dataset*.

Tabela 6.9: Redes que apresentam melhores resultados.

Redes	***Fator de desempenho***					
	Validação			***Teste***		
	RMSE	***PCC***	***SROCC***	***RMSE***	***PCC***	***SROCC***
AlexNet / Tamanho de Minibatch 32 / Imagens compostas em plano	0.63985	**0.9399**	0.9338	1.0495	0.795	0.8435
GoogLeNet / Tamanho de Minibatch 8 / Imagens compostas em plano	**0.52854**	0.9323	0.9326	0.9231	0.8436	0.9068
ResNet-101 / Tamanho de Minibatch 8 / Imagens compostas em plano	0.61465	0.914	**0.9396**	1.0607	0.804	0.8787
AlexNet / Tamanho de Minibatch 32 / Imagens compostas em profundidade	0.97485	0.7923	0.7374	0.5559	0.9703	**0.974**
GoogLeNet / Tamanho de Minibatch 32 / Imagens compostas em profundidade	0.66874	0.8963	0.9045	**0.2067**	**0.9961**	0.9587
ResNet-101 / Tamanho de Minibatch 32 / Imagens compostas em profundidade	0.92916	0.8186	0.8471	0.6026	0.9632	0.9619

6.6 Resultados de *dataset* suplementar de Teste

Depois dos testes com o primeiro *dataset* foi necessário testar as redes com novos conteúdos, de uma forma mais prática e semelhante às condições de teste em aplicações futuras. Para isto foi feito um novo estudo subjetivo descrito na secção 3.5 e com as projeções dessas nuvens foi gerado uma imagem composta em plano e em profundidade para cada PC e cada nível de degradação (um total de 18 por tipo de imagem).

Tabela 6.10: Resultados finais de teste

Redes	***RMSE***	***PCC***	***SROCC***	***OR***
AlexNet / Minibatch 32 / Imagens compostas em plano	1.2274	0.4726	0.6636	0.8889
AlexNet / Minibatch 16 / Imagens compostas em plano	1.2262	0.4697	0.4827	0.7222
AlexNet / Minibatch 8 / Imagens compostas em plano	1.1741	0.5236	0.7359	0.7778
AlexNet / Minibatch 32 / Imagens compostas em profundidade	1.0881	0.6189	0.8	0.8333
AlexNet / Minibatch 16 / Imagens compostas em profundidade	0.5348	0.9713	**0.9137**	0.8333
AlexNet / Minibatch 8 / Imagens compostas em profundidade	0.8613	0.8817	0.8083	0.8333
GoogLeNet / Minibatch 32 / Imagens compostas em plano	1.4578	0.531	0.4868	0.6111
GoogLeNet / Minibatch 16 / Imagens compostas em plano	0.6048	0.9265	0.7308	0.5
GoogLeNet / Minibatch 8 / Imagens compostas em plano	1.1181	0.6728	0.6977	0.7222
GoogLeNet / Minibatch 32 / Imagens compostas em profundidade	**0.2601**	**0.9822**	**0.9137**	0.3889
GoogLeNet / Minibatch 16 / Imagens compostas em profundidade	0.4069	0.9635	0.8724	**0.2778**
GoogLeNet / Minibatch 8 / Imagens compostas em profundidade	0.4979	0.9319	0.9034	0.5
ResNet-101 / Minibatch 32 / Imagens compostas em plano	1.4647	0.113	0.3938	0.8333
ResNet-101 / Minibatch 16 / Imagens compostas em plano	1.4859	0.2985	0.3618	0.7778
ResNet-101 / Minibatch 8 / Imagens compostas em plano	1.4859	0.2714	0.4124	0.8889
ResNet-101 / Minibatch 32 / Imagens compostas em profundidade	0.522	0.9379	0.8186	0.5
ResNet-101 / Minibatch 16 / Imagens compostas em profundidade	0.4566	0.9551	0.8755	0.4444
ResNet-101 / Minibatch 8 / Imagens compostas em profundidade	0.4831	0.9548	0.8506	0.5556

Neste teste houve claramente uma rede que apresentou os melhores resultados, a *GoogLeNet* com um tamanho de *minibatch* de 32 para as imagens compostas em profundidade. Esta rede obteve a melhor raíz do erro médio quadrático (RMSE), o melhor coeficiente de correlação de Pearson (PCC) e o melhor coeficente de correlação de Spearman (SROCC), só não obteve o melhor rácio de *outliers* (OR) sendo que a rede que obteve melhor OR foi a mesma mas para um tamanho de *minibatch* de 16.

Podemos concluir que a *GoogLeNet* é a rede que apresenta melhores resultados na globalidade superando todos os resultados de estado da arte presentes em [17].

Na secção 3.5.3 foram apresentados os fatores de desempenho das principais métricas de avaliação objetiva aplicando um ajuste cúbico aos valores, sendo que o melhor resultado de PCC foi 0.955 para a distância de Hausdorff da métrica ponto-a-plano, o melhor resultado de SROCC foi de 0.8879 para o MSE da métrica ponto-a-ponto, o melhor resultado de RMSE foi de 0.403 para a distância de Hausdorff da métrica ponto-a-plano e o melhor OR foi de 0.444 para a distância de Hausdorff da métrica ponto-a-ponto. Todos estes valores são superados pelos fatores de desempenho da rede *GoogLeNet* com um tamanho de *minibatch* de 32 para as imagens compostas em profundidade, com um resultado de PCC de 0.9822, um resultado de SROCC de 0.9137, um resultado de RMSE de 0.2601 e um resultado de OR de 0.3889.

No gráfico da figura 6.3 estão representados os resultados MOS obtidos no estudo subjetivo referido na secção 5.4 em função dos resultados MOS preditos pela rede *GoogLeNet* com um tamanho de *minibatch* de 32, para as imagens compostas em profundidade de cada tipo de conteúdo do *dataset* suplementar de teste (*Statue Klimt* (losango vermelho), *queen* (triângulo verde) e *loot* (círculo azul)). As barras de erro

representam os intervalos de confiança calculados para os valores MOS, presentes na tabela 5.6.

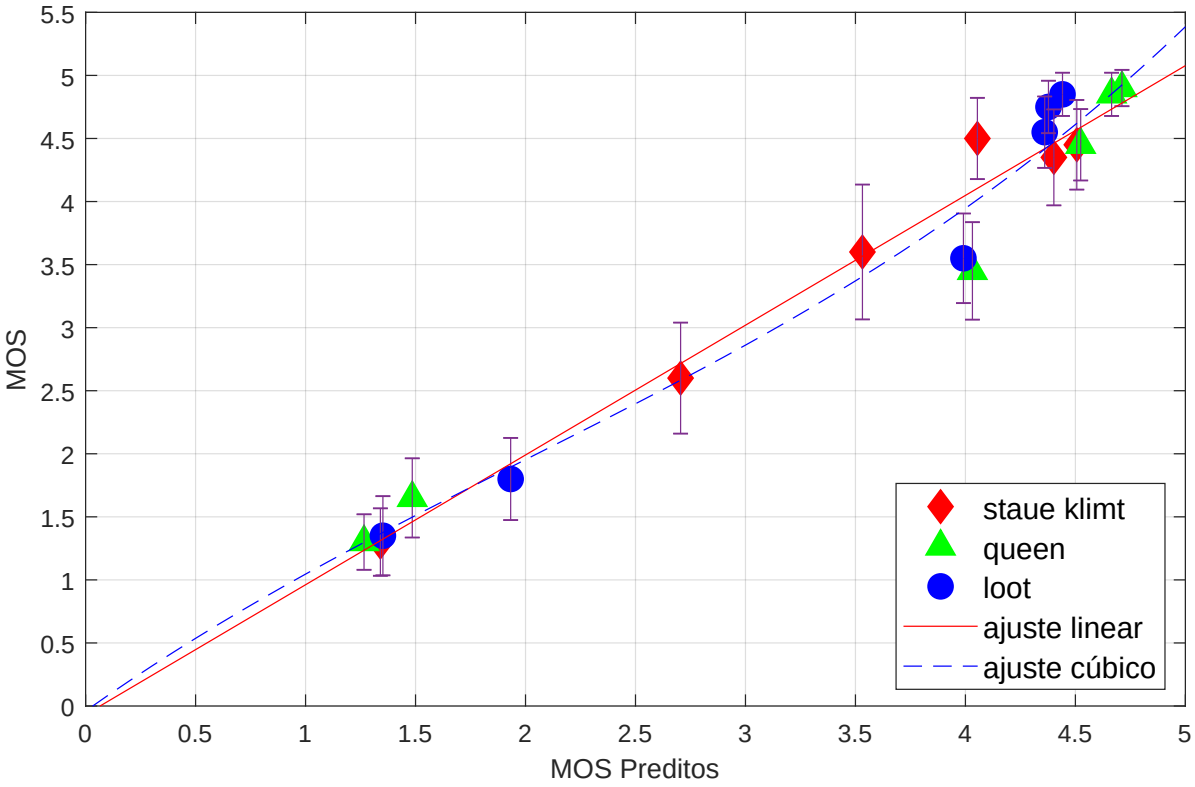


Figura 6.3: Resultados MOS da GoogLeNet / Minibatch 32 / Imagens compostas em profundidade vs resultados MOS subjetivos, com ajustes linear e cúbico.

No gráfico também está presente uma linha vermelha que diz respeito ao ajuste linear feito aos pontos e é representada pela equação 6.1. A linha tracejada a azul diz respeito ao ajuste cúbico feito aos pontos e é representada pela equação 6.2.

$$y_{linear} = 1.029x - 0.0676 \tag{6.1}$$

$$y_{cubico} = 0.0295x^3 - 0.1773x^2 + 1.2328x - 0.0392 \tag{6.2}$$

Por análise da equação 6.1 é possível concluir que o declive da reta é próximo de um, o que indica que os MOS preditos crescem ao mesmo ritmo dos valores MOS reais. Foi também calculado o valor de R^2 e de R^2 *ajustado* sobre o ajuste linear e o ajuste cúbico feitos. Estes valores encontram -se na tabela 6.11.

Tabela 6.11: Valores de R^2 e R^2 ajustado para o ajuste linear e cúbico.

	Ajuste Linear	*Ajuste cúbico*
R^2	0.9646	**0.9684**
R^2 *ajustado*	**0.9624**	0.9616

Pelos valores de R^2 poderíamos ser tentados a afirmar que o modelo cúbico se ajusta melhor aos pontos do que o modelo linear, mas para comparar modelos de ordens diferentes devemos ter em consideração o valor de R^2 ajustado. Como o valor de R^2 ajustado é menor para o ajuste cúbico do que para o ajuste linear,

isto indica que o aumento de complexidade não é justificado. Podemos então concluir que a regressão linear é o modelo que melhor se ajusta ao conjunto de pontos.

A rede *GoogLeNet* com um tamanho de *minibatch* de 32, para as imagens compostas em profundidade depois de treinada demorou 0.245068 segundos a prever os valores MOS para todo o *dataset* suplementar.

7 Conclusão e Trabalho Futuro

Este trabalho apresenta uma nova forma de avaliar qualidade objetiva de nuvens de pontos recorrendo a ferramentas de aprendizagem profunda.

A primeira parte deste trabalho consistiu na criação do *Dataset* Principal tendo por base um estudo de avaliação de qualidade subjetiva [17], desse estudo apenas os conteúdos de pequena escala avaliados (*longdress*, *romanoillamp* e *bumbameuboi* com 3 níveis de degradação para dois métodos de compressão diferentes) fazem parte do *Dataset* Principal.

Este *dataset* foi criado recorrendo às projeções dessas PCs nas faces de um cubo que inclui toda a PC. Isto foi feito para todas as PCs acima descritas para todos os níveis de degradação e para as PCs originais. Para poder aumentar o *dataset* este cubo foi rodado primeiro em torno do eixo x e depois em torno do eixo y em passos de 1° e as projeções foram obtidas para cada passo.

Com todas estas projeções foram criadas imagens compostas que continham as seis faces da PC original e as seis faces da PC degradada, isto para cada conjunto de projeções de cada passo de 1°. A cada imagem composta foi associado o seu valor MOS obtido no estudo [17].

Na segunda parte deste trabalho foi usado este *dataset* para treinar várias CNNs usando 80% para treino e validação, e 20% para teste.

Para o *Dataset* Principal os melhores resultados de validação foram: 0.5285 de RMSE com a rede *GoogLeNet / Tamanho de Minibatch 8 / Imagens compostas em plano*, 0.9399 de PCC com a rede *AlexNet / Tamanho de Minibatch 32 / Imagens compostas em plano* e 0.9396 de SROCC com a rede *ResNet-101 / Tamanho de Minibatch 8 / Imagens compostas em plano*.

Os melhores resultados de teste foram: 0.2067 de RMSE com a rede *GoogLeNet / Tamanho de Minibatch 32 / Imagens compostas em profundidade*, 0.9961 PCC com a mesma rede e 0.974 de SROCC com a rede *AlexNet / Tamanho de Minibatch 32 / Imagens compostas em profundidade*. A partir destes resultados concluiu-se que a rede *GoogLeNet / Tamanho de Minibatch 32 / Imagens compostas em profundidade* foi a que apresentou melhores resultados dando mais peso aos resultados de teste.

Por fim a terceira parte deste trabalho foi a realização de um Teste Suplementar de avaliação subjetiva de qualidade descrito nas secções 3.5, 5.2 e 5.4, para poder testar as redes com um novo *dataset* e com PCs nunca antes vistas pelas redes.

Os melhores resultados obtidos para estes conteúdos foram: 0.2601 de RMSE para a rede *GoogLeNet / Tamanho de Minibatch 32 / Imagens compostas em profundidade*, 0.9822 de PCC para mesma rede, 0.9137 de SROCC também para a mesma rede e 0.2778 de OR para a rede *GoogLeNet / Tamanho de Minibatch 16 / Imagens compostas em profundidade*. Com estes resultados foi possível concluir que a rede que atinge os melhores resultados é a *GoogLeNet / Tamanho de Minibatch 32 / Imagens compostas em profundidade*,

resultados esses que superam os resultados de estado da arte de avaliação de qualidade objetiva de PCs com textura presentes em [17], e todos os resultados calculados na secção 3.5.3 dos fatores de desempenho das principais métricas de avaliação objetiva de PCs para as nuvens do *dataset* suplementar.

Neste trabalho foram usados dois *datasets* bastante reduzidos e com níveis de degradação específicos, apenas usando dois métodos de compressão diferentes. Para trabalho futuro esta rede poderá ser treinada com mais conteúdos utilizando mais métodos de compressão com mais níveis de degradação. Para isso terão de ser feitos mais estudos de avaliação subjetiva de qualidade de PCs.

Outra limitação deste trabalho é o facto da necessidade da obtenção das projeções das PCs e toda uma organização destas projeções em agregados de imagens para entrada da rede, para trabalho futuro um dos possíveis desafios poderá ser utilizar diretamente as PCs como entrada da CNN.

8 Bibliografia

[1] Cisco, Visual Network Index, "Forecast and methodology, 2017-2027," *White Paper*, 2018.

[2] A. Géron, "Hands-on machine learning with scikit-learn and tensorflow: concepts, tools, and techniques to build intelligent systems," *O'Reilly Media, Inc.*, 2017.

[3] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for no-reference image quality assessment," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1733–1740, 2014.

[4] S. Bosse, D. Maniry, K.-R. Müller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 206–219, 2018.

[5] F. Pereira and E. A. B. da Silva, "Efficient plenoptic imaging representation: Why do we need it?," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, July 2016.

[6] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006.

[7] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1–4, May 2011.

[8] Z. Li, B. Barsky, and X. Jin, "An effective third-order local fitting patch and its application," in *2009 IEEE International Conference on Shape Modeling and Applications*, pp. 7–14, IEEE, 2009.

[9] Z. Wang and A. C. Bovik, "Modern image quality assessment," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 2, no. 1, pp. 1–156, 2006.

[10] E. Dumic, C. R. Duarte, and L. A. da Silva Cruz, "Subjective evaluation and objective measures for point clouds — state of the art," in *2018 First International Colloquium on Smart Grid Metrology (SmaGriMet)*, pp. 1–5, April 2018.

[11] R. N. Mekuria, Z. Li, C. Tulvan, and P. Chao, "Evaluation criteria for PCC (point cloud compression)," *ISO/IEC JTC1/SC29/WG11 MPEG2016/n16332*, 2016.

[12] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Geometric distortion metrics for point cloud compression," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3460–3464, Sept 2017.

[13] E. Alexiou and T. Ebrahimi, "Point cloud quality assessment metric based on angular similarity," in *International Conference on Multimedia and Expo (ICME)*, no. CONF, 2018.

[14] E. M. Torlig, E. Alexiou, T. A. Fonseca, R. L. de Queiroz, and T. Ebrahimi, "A novel methodology for quality assessment of voxelized point clouds," in *Applications of Digital Image Processing XLI*, vol. 10752, p. 107520I, International Society for Optics and Photonics, 2018.

[15] E. M. Torlig, "Métricas objetivas baseadas em projeções para avaliação de qualidade em nuvens de pontos," Tese de Mestrado em Engenharia Elétrica, Universidade de Brasília, Brasil, 2018.

[16] ITU-R BT.500-13, "Methodology for the subjective assessment of the quality of television pictures," in *International Telecommunications Union*, January 2012.

[17] L. A. Silva Cruz, E. Dumic, E. Alexiou, J. Prazeres, R. Duarte, M. Pereira, A. Pinheiro, and T. Ebrahimi, "Point cloud quality evaluation : Towards a definition for test conditions," in *11th International Conference on Quality of Multimedia Experience (QoMEX)*, no. CONF, 2019.

[18] JPEG Pleno Database, https://jpeg.org/plenodb/.

[19] 3DTK – The 3D Tookit, http://threedtk.de.

[20] "3D point cloud and mesh processing software Open Source Project."

[21] ITU-T P.1401, "Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models," in *International Telecommunications Union*, July 2012.

[22] E. Alexiou, M. V. Bernardo, L. A. da Silva Cruz, L. G. Dmitrovic, C. Duarte, E. Dumic, T. Ebrahimi, D. Matkovic, M. Pereira, A. Pinheiro, *et al.*, "Point cloud subjective evaluation methodology based on 2d rendering," in *10th International Conference on Quality of Multimedia Experience (QoMEX)*, no. CONF, 2018.

[23] S. Albawi, T. Abed MOHAMMED, and S. ALZAWI, "Understanding of a convolutional neural network," 08 2017.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.