

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Arthur Henrique de Souza

**Definição de parâmetros de iluminação de
cenais virtuais 3D a partir de imagens**

Uberlândia, Brasil

2026

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Arthur Henrique de Souza

**Definição de parâmetros de iluminação de cenas virtuais
3D a partir de imagens**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, como parte dos requi-
sitos exigidos para a obtenção título de Ba-
charel em Ciência da Computação.

Orientador: Bruno Augusto Nassif Travençolo

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2026

Resumo

A iluminação realista de ambientes virtuais 3D exige configurações complexas e alto domínio técnico. Este trabalho propõe o desenvolvimento e a integração de um sistema baseado em inteligência artificial para automatizar a parametrização fotorrealista da iluminação em motores gráficos, utilizando apenas uma fotografia panorâmica como entrada. A metodologia consiste no treinamento de três Redes Neurais Convolucionais (CNNs) baseadas na arquitetura AlexNet. O conjunto de dados original, construído a partir das bases Laval Outdoor e Poly Haven, foi pré-processado com técnicas de aumento de dados e discretização angular. Duas redes operam como classificadoras para estimar as coordenadas esféricas do sol (zênite e azimuth), enquanto a terceira atua como modelo de regressão multivariada para extrair propriedades atmosféricas fundamentadas no modelo de Lalonde-Matthews. O sistema foi integrado nativamente na *Unity Game Engine 6* por meio da biblioteca Unity Sentis, utilizando modelos exportados no formato ONNX. As avaliações demonstraram alta precisão na predição do zênite (erro médio de 1,30 classes) e dos parâmetros difusos (Erro Absoluto Médio global de 13,87%), embora a rede de azimuth tenha enfrentado desafios de predição devido a ambiguidades visuais no horizonte. Qualitativamente, o sistema conseguiu traduzir as saídas matemáticas em propriedades visuais coerentes, modulando a *Directional Light* e o *Skybox* para reproduzir fielmente a coloração e a densidade luminosa da imagem original. Conclui-se que o método automatiza a estruturação inicial da iluminação, acelerando significativamente o fluxo de trabalho de desenvolvimento de jogos e simulações, fornecendo uma base robusta que requer apenas pequenos ajustes finos humanos para alcançar o fotorrealismo.

Palavras-chave: Iluminação virtual, redes neurais convolucionais, visão computacional, Unity Sentis, Unity.

Lista de ilustrações

Figura 1 – Fluxo de utilização da ferramenta integrada na <i>Unity</i> : (A) Cena inicial; (B) Configuração da inferência; (C) Cena final atualizada.	7
Figura 2 – Comparação entre funções de ativação, mapeando a entrada no eixo x para a respectiva saída no eixo y	9
Figura 3 – Imagem retirada do site (Unity Technologies, 2026a) que demonstra o <i>Universal Render Pipeline (URP)</i> numa cena virtual na Unity3D.	10
Figura 4 – Imagem retirada do site (Unity Technologies, 2026a) que demonstra o HDRP numa cena virtual na Unity3D.	11
Figura 5 – Imagens adaptadas do site (Unity Technologies, 2026f) que mostram cada corpo de luz primário na Unity3D.	12
Figura 6 – Imagem retirada do site (Unity Technologies, 2026b) que demonstra o volume <i>View Frustum</i>	12
Figura 7 – Alterando a posição e rotação da câmera a visualização da cena é alterada.	13
Figura 8 – Imagem retirada do artigo (IORNS; RHEE, 2015) que ilustra o a iluminação a partir de imagens.	15
Figura 9 – Imagem retirada do artigo (BHATTAD; SOOLE; FORSYTH, 2024) que mostra o resultado do processo de re-iluminação.	15
Figura 10 – Resultado da configuração de luz automática.	16
Figura 11 – Diagrama de blocos ilustrando as quatro etapas principais da metodologia proposta.	19
Figura 12 – Arquitetura da rede neural que encontra o azimuth do sol	22
Figura 13 – Arquitetura da rede neural que encontra o zênite do sol	23
Figura 14 – Captura mostrando os parâmetros do componente Sentis Sky Controller	25
Figura 15 – Diagrama que demonstra a arquitetura da fase de treinamento da rede neural e da implementação na unity	28
Figura 16 – Recorte utilizado para avaliar o modelo.	31
Figura 17 – Renderização da cena utilizando os parâmetros de iluminação reais (<i>Ground Truth</i>).	31
Figura 18 – Renderização da cena utilizando os parâmetros de iluminação estimados pela rede neural.	32

Lista de tabelas

Tabela 1 – Resultados da avaliação das redes neurais de classificação espacial do sol.	29
Tabela 2 – Erro Absoluto Médio (MAE) para os parâmetros de iluminação de Lalonde-Matthews.	30

Sumário

1	INTRODUÇÃO	6
2	FUNDAMENTAÇÃO TEÓRICA	8
2.1	Redes Neurais Convolucionais	8
2.2	Unity3D	9
2.3	Modelos de Iluminação e Câmera Virtual	10
2.4	Python Keras	12
2.5	OpenCV	13
2.6	Unity Sentis	13
3	TRABALHOS RELACIONADOS	14
4	DESENVOLVIMENTO	18
4.1	Base de Dados	18
4.2	Implementação	20
4.3	Redes Neurais	21
5	RESULTADOS	29
5.1	Avaliação das Redes de Posição Solar (Zênite e Azimute)	29
5.2	Avaliação da Rede de Parâmetros de Iluminação	30
6	CONCLUSÃO	33
	REFERÊNCIAS	35

1 Introdução

O uso de cenários virtuais 3D é recorrente nos filmes, nos ambientes de realidade virtual e nos jogos eletrônicos. Os jogos digitais são uma indústria bem estabelecida, chegando a faturar US\$ 175,8 bilhões em 2021 de acordo com o site ([NEWZOO, 2022](#)). É usual que esses ambientes busquem ser realistas e, com isso, a luz deve ser simulada com um alto grau de precisão. A iluminação virtual é uma simulação com uma grande gama de parâmetros, muitas vezes sendo necessário conhecimento profundo em computação gráfica para compreender melhor as opções de parametrização. Portanto, o uso de uma imagem real como base para a configuração da iluminação simulada pode agilizar o desenvolvimento dessas mídias.

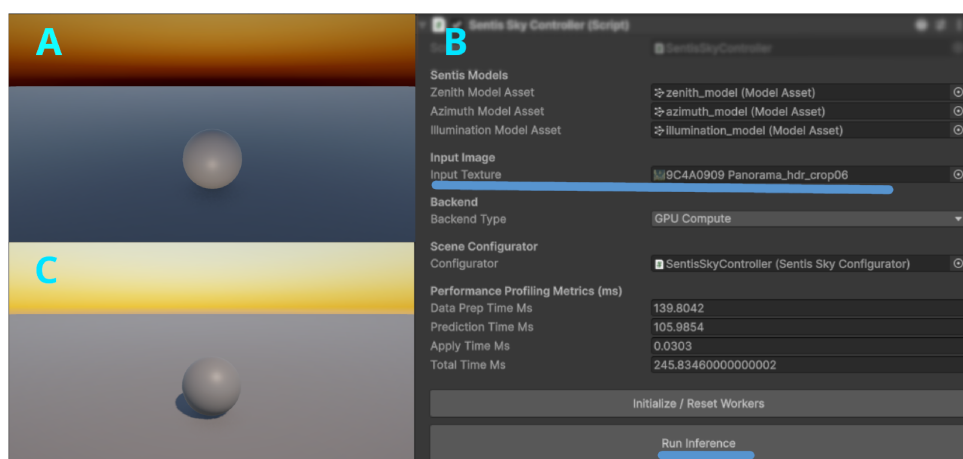
A ideia principal explorada neste trabalho é, a partir dos pixels de uma imagem panorâmica, obter dados sobre a iluminação da cena, como a posição do sol, a intensidade da luz e outros parâmetros. Com esses dados é possível realizar a configuração de esquemas de iluminação de um ambiente virtual 3D e, com isso, simular sombras e reflexões de acordo com uma fonte de luz virtual.

A Figura 1 mostra um exemplo de como funciona o processo da ferramenta. Inicialmente, em (A), apresenta-se uma cena 3D básica no motor gráfico *Unity*, composta por um plano e uma esfera, sob uma iluminação global padrão. Na etapa (B), o usuário configura o componente *Sentis Sky Controller* por meio do inspetor, atribuindo a imagem desejada como entrada (*Input Texture*) e acionando o botão de inferência. Por fim, em (C), o resultado é aplicado automaticamente à cena, evidenciando uma alteração drástica na iluminação global e na projeção de sombras.

O trabalho de conclusão de curso ([BERWALDT, 2021](#)), usado como base para essa pesquisa, conseguiu, com o uso de redes neurais, um resultado satisfatório na configuração da iluminação de um ambiente 3D.

A informação sobre a iluminação na imagem de entrada será extraída pelo modelo Lalonde e Matthews. Para simular a iluminação em um ambiente 3D virtual será usada a *Unity Game Engine*, uma ferramenta gratuita e de fácil uso. Com o intuito de fazer a interface entre a *engine* e o modelo neural, vai ser usado o pacote *Sentis*. Além disso, a interface de programação *Keras* será usada para o treinamento da rede e *OpenCV* para manipulação de imagens, sendo ambas bibliotecas disponíveis na linguagem de programação *Python*.

Figura 1 – Fluxo de utilização da ferramenta integrada na *Unity*: (A) Cena inicial; (B) Configuração da inferência; (C) Cena final atualizada.



Fonte: Elaborada pelo autor.

2 Fundamentação teórica

Nesta seção serão apresentados os principais conceitos e ferramentas necessários para a compreensão deste trabalho. Serão abordadas as Redes Neurais Convolucionais, o motor gráfico Unity e, por fim, as bibliotecas Python Keras, OpenCV e Unity Sentis.

2.1 Redes Neurais Convolucionais

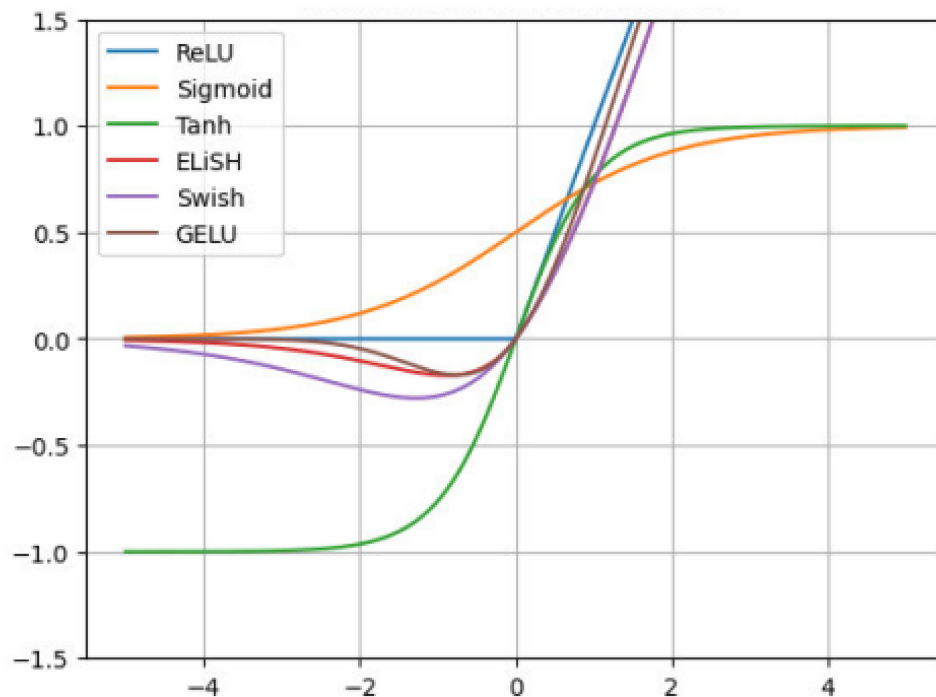
O trabalho (KHAN et al., 2020) define redes neurais convolucionais como um sistema de inteligência artificial que é capaz de realizar uma análise profunda em uma imagem, utilizando múltiplas camadas de neurônios simulados, com isso conseguindo aprender apenas com os dados de entrada, não sendo mandatória a intervenção humana durante o processo de aprendizagem.

O artigo procede descrevendo os diversos tipos de camadas que uma rede neural convolucional geralmente possui; dentre elas, está a camada de convolução, que oferece filtros a serem aplicados diversas vezes para extrair informações da imagem de entrada por meio de operações matriciais. Ainda no contexto das redes neurais, a publicação (PUSZ-TAHÁZI; EIGNER; CSISZÁR, 2024) entra em detalhes sobre a camada de ativação, que de acordo com uma função não linear aplicada à entrada, vai definir se o neurônio deve ou não ser ativado. Algumas funções comumente usadas nessa camada são: Sigmoid, que recebe um valor real como entrada e como saída devolve um valor entre 0 e 1; Tanh, que também recebe um valor real e retorna um valor entre -1 e 1; ReLU, ao receber um valor de entrada X , ele retornará $\max(0, X)$. A Figura 2 ilustra as funções citadas.

Além disso, o estudo (KHAN et al., 2020) apresenta uma tabela descrevendo os principais tipos de arquitetura de redes neurais convolucionais, dentre elas estão:

- **LeNet (1998)**: foi a primeira arquitetura popular de uma rede neural convolucional, com 7 camadas. É eficaz para fazer o processamento de imagens de entrada de baixa escala;
- **AlexNet (2012)**: com suas 8 camadas, possui uma maior amplitude e capacidade de processamento.
- **DenseNet-121 (2017)**: com mais de 190 camadas, cada uma conectada com a outra, ela diminui a ocorrência de certos problemas de uma CNN, como o desaparecimento de gradientes.

Figura 2 – Comparação entre funções de ativação, mapeando a entrada no eixo x para a respectiva saída no eixo y .



Fonte: (PUSZTAHÁZI; EIGNER; CSISZÁR, 2024).

A publicação compila informações sobre as redes neurais convolucionais; dentre elas, estão inovações recentes da área, como o mecanismo de atenção, que incrementa a performance de uma CNN, permitindo que o modelo tenha uma região de foco.

2.2 Unity3D

De acordo com o site (Unity Technologies, 2026c), *Unity3D* é um motor gráfico especializado no desenvolvimento de jogos, facilitando a renderização e iluminação de cenas virtuais, provendo diversos recursos para a simulação de física, a manipulação de *sprites* e outras facilidades.

A etapa inicial na criação de um novo projeto na *Unity3D* é escolher qual *render pipeline* será usado no jogo. A página de internet (Unity Technologies, 2026a) define como o motor usa o *pipeline* para renderizar um quadro de uma cena virtual:

1. *Culling*, quando é escolhido quais objetos serão mostrados na tela de fato. Objetos fora do alcance da câmera virtual são removidos, e objetos que estão atrás de outros podem ser parcialmente mostrados ou completamente escondidos.
2. *Rendering*, quando o pipeline desenha o objeto na tela com as cores definidas pelo processo de iluminação.

3. *Post-processing*, o pipeline aplica modificações como correção de cor, *bloom* e outros efeitos na imagem renderizada no passo anterior.

O motor gráfico oferece três *render pipelines*; cada um deles possui um foco específico, devendo ser escolhido cuidadosamente, já que cada pipeline oferece diferentes recursos de iluminação, desempenho, plataformas suportadas, customização e outras diferenças.

De acordo com a página (Unity Technologies, 2026d) e a (Unity Technologies, 2026a), os três *pipelines* são: (i) *Universal Render Pipeline (URP)*, possuindo uma vasta gama de plataformas que conseguem executar esse *pipeline*, e diversas opções de customizações de fácil acesso, podendo ser usado por usuários de qualquer nível técnico. A Figura 3 mostra como o URP pode ser usado; (ii) *High Definition Render Pipeline (HDRP)*: o foco dele é ser fotorrealista, então apenas plataformas de alto desempenho conseguem executá-lo. Possui diversos recursos avançados como *Ray Tracing*, *Path Tracing* e outras tecnologias de ponta para renderizar a cena de forma mais realista possível. A Figura 4 mostra como que o HDRP pode ser usado para atingir o fotorrealismo; e (iii) *Built-In Render Pipeline* é o pipeline com menos opções de customização, devendo ser usado apenas para casos simples.

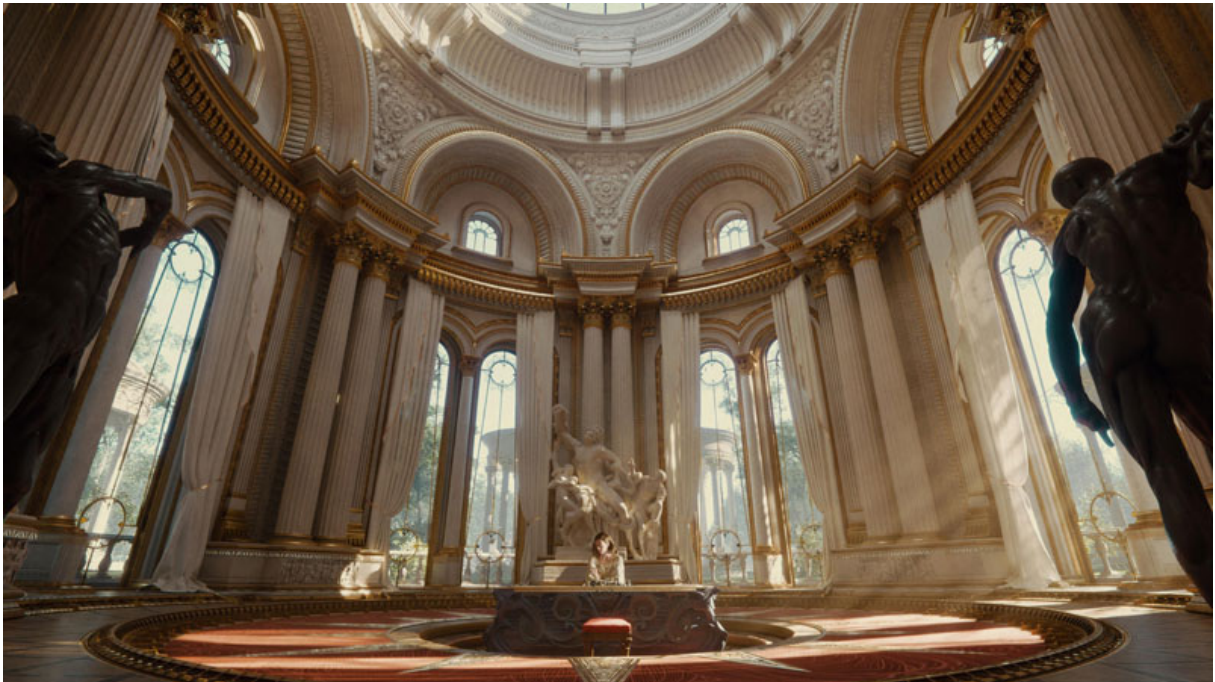
Figura 3 – Imagem retirada do site (Unity Technologies, 2026a) que demonstra o *Universal Render Pipeline (URP)* numa cena virtual na Unity3D.



2.3 Modelos de Iluminação e Câmera Virtual

A página (Unity Technologies, 2026f) define como pode ser feita a iluminação de uma cena virtual com o uso de quatro objetos de luz primária, ilustrados na Figura 5:

Figura 4 – Imagem retirada do site ([Unity Technologies, 2026a](#)) que demonstra o **HDRP** numa cena virtual na Unity3D.



- *Point Light* é posicionada no cenário emitindo uma luz em um formato esférico; quanto mais próxima a luz, mais intensa a luz será refletida no objeto.
- *Spot Light* também é posicionada no cenário, mas emite a luz em formato de um cone.
- *Directional Light* emite a luz em uma direção e é comumente usada para simular uma grande fonte de luz distante, como o sol.
- *Area Light* é posicionada no cenário virtual, emitindo luz uniformemente em uma área 2D.

Além da iluminação, a visualização e a renderização do ambiente tridimensional dependem diretamente da Câmera Virtual. De acordo com a *Unity* ([Unity Technologies, 2026b](#)), o espaço físico que a câmera consegue capturar é matematicamente representado por um volume denominado *View Frustum*. A Figura 6 mostra que esse volume possui o formato de uma pirâmide truncada e é delimitado pelo ângulo de abertura da lente, chamado de Campo de Visão, e por dois planos paralelos: o *Near Clip Plane* e o *Far Clip Plane*. Apenas a geometria presente dentro deste tronco é processada e exibida pelo motor gráfico. Conforme mostra a Figura 7, quando a câmera se move e rotaciona, o tronco é atualizado, alterando assim a perspectiva exibida ao usuário.

Figura 5 – Imagens adaptadas do site ([Unity Technologies, 2026f](#)) que mostram cada corpo de luz primário na Unity3D.

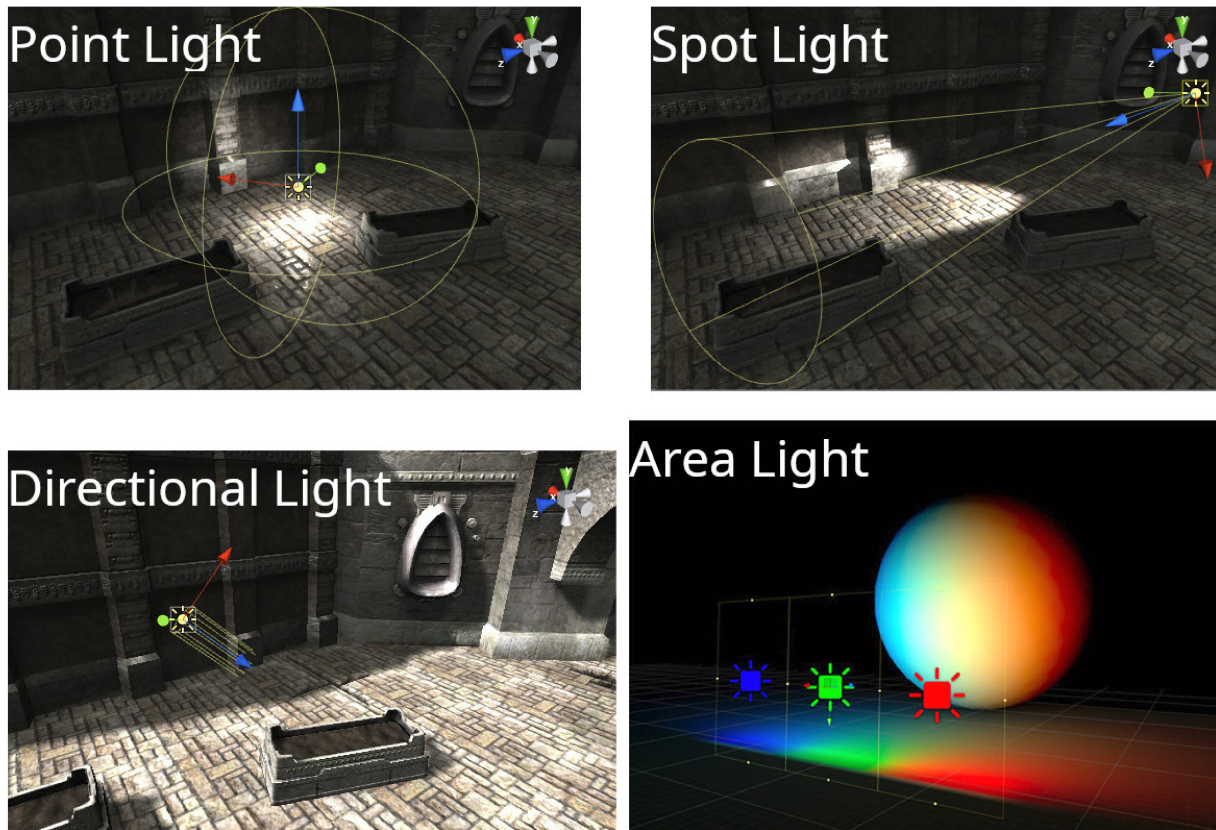
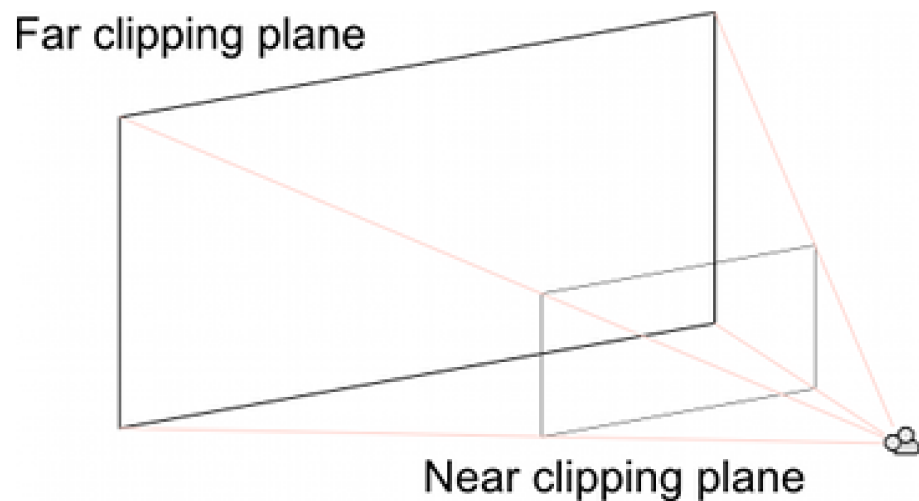


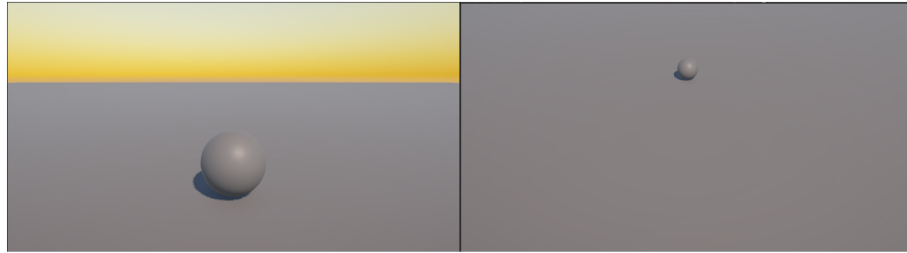
Figura 6 – Imagem retirada do site ([Unity Technologies, 2026b](#)) que demonstra o volume *View Frustum*.



2.4 Python Keras

A biblioteca Python Keras 3 oferece uma API *multi-framework* voltada ao aprendizado profundo (*deep learning*). Conforme evidenciado em sua documentação oficial ([CHOLLET, 2015](#)), o Keras permite que o desenvolvedor utilize JAX, TensorFlow ou PyTorch como *backend*, sem a necessidade de um aprofundamento específico em cada um

Figura 7 – Alterando a posição e rotação da câmera a visualização da cena é alterada.



Fonte: Elaborada pelo autor.

desses ecossistemas. Essa característica possibilita a criação de componentes modulares que mantêm a compatibilidade entre os diferentes *frameworks*.

2.5 OpenCV

A biblioteca OpenCV (*Open Source Computer Vision Library*) é definida como uma solução de código aberto para visão computacional e processamento de imagens (OpenCV, 2026). Ela dispõe de mais de 2500 algoritmos otimizados para diversas tarefas, incluindo a identificação de objetos e faces, a extração de modelos 3D e outras funções essenciais para a análise visual automatizada.

2.6 Unity Sentis

O Unity Sentis, por sua vez, é caracterizado como uma interface para a execução de redes neurais no motor gráfico (Unity Technologies, 2026e). O pacote viabiliza a importação de modelos pré-treinados para o ambiente da Unity, permitindo sua execução em tempo real durante a aplicação. O sistema oferece suporte a formatos estabelecidos na indústria, como *Open Neural Network Exchange* (ONNX), *LiteRT* e modelos nativos do PyTorch.

3 Trabalhos relacionados

Em 2011 houve um avanço na área de iluminação baseada em imagem, devido ao trabalho de (CHEN; WANG; JIN, 2011). Nesse estudo, foram utilizados parâmetros de textura, bordas e profundidade para fazer uma análise geométrica da imagem, além da análise de sombra e reflexão. Com isso, é possível adicionar um objeto virtual à imagem e simular como a luz afeta esse objeto.

No ano seguinte, o trabalho de (LALONDE; EFROS; NARASIMHAN, 2012) propôs o uso de imagens comuns para análise do céu e sombras como pistas para fazer uma estimativa eficaz sobre os parâmetros de iluminação.

O estudo de (IORNS; RHEE, 2015) conseguiu um grande avanço ao demonstrar a renderização de um objeto 3D tendo como base um vídeo panorâmico. Na Figura 8 são mostradas várias aplicações da técnica. A renderização ocorre em tempo real, e a única fonte de informação de iluminação é a do vídeo. Com o progresso das pesquisas, em 2019 (HOLD-GEOFFROY; ATHAWALE; LALONDE, 2019) propuseram o uso de redes neurais convolucionais para estimar a iluminação em uma imagem.

A recente publicação (MIRBAUER et al., 2024) aborda a geração de imagens realistas do céu com nuvens. A técnica proposta permite que o usuário controle a posição do sol e a cobertura de nuvens. Logo, a imagem gerada poderia servir como base para a renderização de uma cena virtual.

No mesmo ano, (BHATTAD; SOOLE; FORSYTH, 2024) propõem um método de re-iluminação de imagens, modificando-as com sombras, reflexos e alguns efeitos de luz. O método consiste em decompor a imagem e encontrar o espaço latente no StyleGAN, um modelo de geração de imagens realista, pré-treinado para identificar assim as direções da luz. Com isso, é possível re-iluminar a cena conforme o desejo do usuário. A Figura 9 ilustra o resultado de diversos processos de re-iluminação aplicados em diferentes imagens.

A monografia (BERWALDT, 2021) tem como objetivo configurar de forma automática a iluminação de uma cena 3D no motor gráfico Unity, atingindo tal resultado utilizando uma imagem de entrada como base de iluminação. O trabalho consegue realizar tal feito usando uma rede neural convolucional para estimar os parâmetros de iluminação da imagem de entrada. A rede foi treinada com a base de dados Laval Outdoor, que possui 205 imagens panorâmicas de espaços abertos. Além disso, o modelo de representação de iluminação adotado é o apresentado no trabalho (LALONDE; MATTHEWS, 2014), que possui uma quantidade de parâmetros de iluminação significativa sem pesar o processamento de extração de parâmetros da imagem. O estudo (BERWALDT, 2021) apresenta um resultado satisfatório ao comparar visualmente a iluminação da imagem de entrada

Figura 8 – Imagem retirada do artigo (IORNS; RHEE, 2015) que ilustra o a iluminação a partir de imagens.



Figura 9 – Imagem retirada do artigo (BHATTAD; SOOLE; FORSYTH, 2024) que mostra o resultado do processo de re-iluminação.

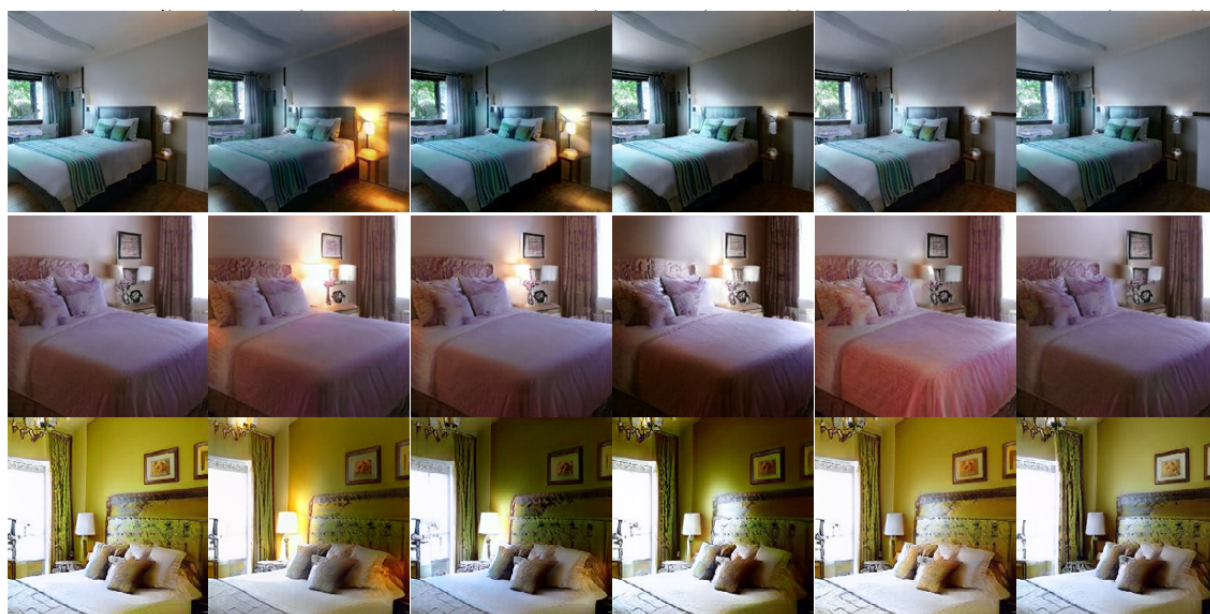


Figura 10 – Resultado da configuração de luz automática.



Figura obtida de (BERWALDT, 2021).

com a iluminação da cena 3D obtida de forma automática. Há uma maior diferença na iluminação se a imagem de entrada possuir algum objeto bloqueando o sol, mas ainda assim os resultados são satisfatórios, como mostrado na Figura 10.

O artigo (IORNS; RHEE, 2015) apresenta uma forma de usar vídeos panorâmicos como parâmetros de iluminação em uma cena 3D renderizada em tempo real. O estudo consegue esse feito fazendo o uso de vídeos que são reconstruídos em mapas de radiância difusa de baixa resolução para acelerar a renderização difusa; consoante a isso, há um esquema de amostragem especular baseado em mipmap que é utilizado para proporcionar uma renderização eficiente na placa de vídeo. O sistema citado é capaz de renderizar em tempo real objetos em uma cena 3D, isso em alta resolução e quadros por segundo e com a iluminação virtual sendo baseada em um vídeo, como mostrado na Figura 8.

O estudo (CHOI; KANG; YUN, 2020) apresenta uma abordagem para estimar a iluminação em imagens utilizando redes neurais convolucionais (CNNs) que incorporam informações semânticas. Os autores propõem uma rede neural convolucional em que, durante o seu treinamento, as imagens de entrada são divididas em frações para extrair informações locais; posteriormente, as informações coletadas são processadas numa camada de *pooling* semântico, gerando um mapa de pesos. Após isso, as informações locais são combinadas, gerando uma estimativa da informação global, com a rede sendo capaz de detectar quais dados são úteis ou ruidosos. A rede demonstrou robustez, operando de forma eficaz em diferentes condições de iluminação, ângulo da imagem e outras variações;

além disso, ela foi comparada a outros métodos de estimação de luz e teve um desempenho superior na maior parte dos casos.

4 Desenvolvimento

Neste capítulo será apresentado o método desenvolvido. O fluxo de trabalho consiste em quatro etapas principais: (i) Definição e pré-processamento da base de dados; (ii) Configuração do ambiente de implementação; (iii) Modelagem e treinamento das redes neurais; e (iv) Integração do modelo de inteligência artificial no motor gráfico Unity.

O diagrama da Figura 11 ilustra a relação e a ordem de execução de cada um desses passos.

4.1 Base de Dados

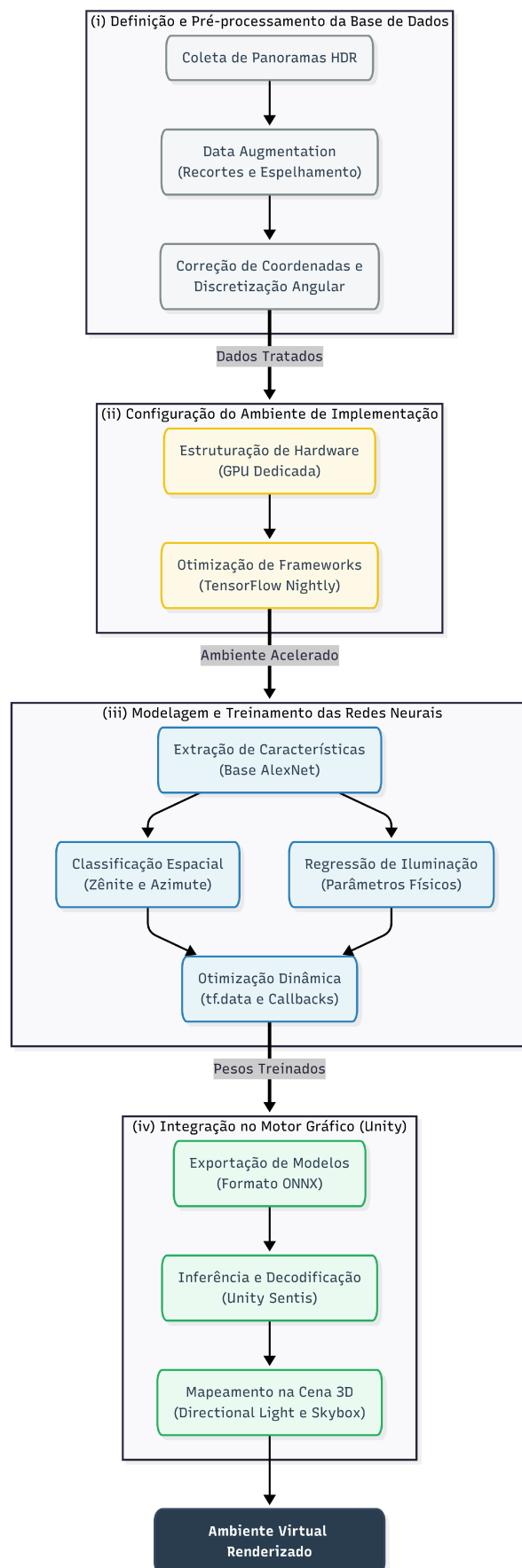
Antes da implementação das redes neurais, foi necessária a estruturação de um banco de dados de imagens de ambientes externos para a fase de treinamento. As imagens precisavam ser panorâmicas em 360 graus, com o sol visível, possibilitando a identificação da origem da emissão luminosa e a análise de como ela está influenciando o restante da cena presente na imagem. Adicionalmente, o formato panorâmico viabilizou a extração de múltiplos recortes a partir de uma única imagem, técnica empregada para aumentar o volume de amostras do conjunto de treinamento.

A base de dados Laval Outdoor Sky Database ([LALONDE, 2019](#)), do professor Jean-François Lalonde, foi fundamental: além de possuir 205 panoramas HDR (*High Dynamic Range*), o conjunto possui anotações da posição do sol e dos parâmetros de iluminação presentes na imagem, de acordo com o modelo de Lalonde-Matthews. Nesse conjunto e também nesse trabalho, a posição do sol será definida por coordenadas esféricas: zênite e azimuth. Neste trabalho, assim como no referido conjunto de dados, a posição solar é representada por meio de coordenadas esféricas compostas pelo ângulo zênite, a elevação angular do Sol em relação à vertical do observador, e pelo azimuth, o ângulo no plano horizontal medido a partir do Norte geográfico.

Além desse conjunto, foram utilizados 30 panoramas do website Poly Haven ([Poly Haven, 2026](#)). Como essas imagens não possuíam os parâmetros de iluminação nem a posição do sol anotados, foi implementada uma ferramenta em *Python* na qual o usuário pode marcar, na imagem, onde está o sol ao clicar com o mouse, salvando assim as coordenadas para uso posterior. Desta forma, esse pequeno conjunto pode ser usado para o treinamento das redes neurais que estimam a posição do sol.

O volume inicial de dados mostrou-se insuficiente para atingir a convergência adequada do modelo, especialmente para a estimativa do ângulo do azimuth da posição solar, o que tornou necessária a aplicação de técnicas de recorte e aumento de dados. Para

Figura 11 – Diagrama de blocos ilustrando as quatro etapas principais da metodologia proposta.



tanto, simulou-se o posicionamento de uma câmera virtual no centro da imagem panorâmica original, com um campo de visão configurado em 80° ; foram geradas 15 novas imagens a partir de cada fotografia original.

Essa amostragem foi obtida através da rotação da câmera em cinco ângulos horizontais (azimute) espaçados simetricamente ($0^\circ, 72^\circ, 144^\circ, 216^\circ, 288^\circ$) combinados a três ângulos de elevação vertical ($-20^\circ, 0^\circ, +20^\circ$), variando assim a altura do horizonte. Para evitar distorções nas bordas dos recortes, inerentes à geometria da imagem esférica, aplicou-se a projeção retilinear durante a extração das perspectivas.

Devido à rotação da câmera virtual, a posição do sol em cada recorte precisou ser recalculada. O sistema realiza a leitura da posição solar normalizada no conjunto original, converte os valores para radianos e aplica o ajuste no azimute conforme o deslocamento horizontal da câmera:

$$\theta_{novo} = \theta_{sol} - \theta_{camera}$$

A rede neural prevê que o azimute esteja entre o intervalo $[-\pi, \pi]$; para garantir isso, após a correção do azimute, é necessário realizar a normalização:

$$\theta_{novo} = ((\theta + \pi) \bmod (2\pi)) - \pi$$

As coordenadas solares são discretizadas para modelar o problema como classificação. No trabalho (BERWALDT, 2021), isso resultou em uma melhor estabilidade no treinamento da rede neural, melhorando os resultados de predição. O valor de azimute é discretizado em 32 classes, e o valor de zênite é discretizado em 16 intervalos.

Adicionalmente, aplicou-se o espelhamento horizontal em todos os recortes, duplicando o volume do banco de dados. Essa técnica demandou uma adaptação complementar no azimute, visto que a inversão horizontal da imagem altera a posição relativa da fonte luminosa em relação ao centro do quadro (uma fonte posicionada à direita na imagem original passará a figurar à esquerda na versão espelhada).

4.2 Implementação

Um dos desafios iniciais do projeto consistiu na configuração do ambiente de desenvolvimento computacional. A estação de trabalho utilizada baseia-se no sistema operacional Linux (distribuição Fedora KDE 43) e é equipada com uma unidade de processamento gráfico (GPU) dedicada de arquitetura recente, o modelo NVIDIA GeForce RTX 5060 Ti.

Essa combinação de *hardware* e *software* gerou incompatibilidades iniciais com as bibliotecas Keras e TensorFlow, resultando no não reconhecimento da GPU. Consequentemente, as etapas de treinamento da rede neural passaram a ser processadas exclusivamente pela Unidade Central de Processamento (CPU), o que acarretou um aumento substancial no tempo computacional exigido para a conclusão de cada época de treinamento.

Após uma série de testes de ambiente, a solução viável encontrada foi a adoção da versão *nightly* do TensorFlow, uma ramificação de desenvolvimento prévio que recebe atualizações diárias. Essa versão específica contemplava o suporte atualizado aos *drivers* da nova placa de vídeo, permitindo a aceleração via *hardware* e, consequentemente, otimizando de forma significativa o tempo de desenvolvimento da pesquisa.

4.3 Redes Neurais

Para o progresso deste estudo, foram implementadas três redes neurais convolucionais baseadas na arquitetura AlexNet. A primeira é dedicada à estimativa do ângulo de zênite solar, a segunda, à predição do azimuth, e a terceira tem a finalidade de extrair os parâmetros de iluminação da cena, fundamentando-se no modelo de Lalonde-Matthews. Todas as redes recebem como entrada imagens padronizadas com dimensões de 128 x 256 pixels.

Focando especificamente nos modelos responsáveis por estimar a posição espacial do sol, ambos compartilham a mesma estrutura base de extração de características. Essa base é composta por cinco blocos convolucionais, responsáveis por mapear padrões visuais hierárquicos, como bordas, cores e formas, seguidos por duas camadas densas, que atuam na interpretação unificada dessas características. Visando mitigar o problema de *overfitting*, aplicou-se a técnica de *Dropout* com uma taxa de 50% nas camadas densas finais, garantindo que o modelo generalize melhor os dados em vez de apenas memorizar o conjunto de treinamento.

A distinção arquitetural entre essas duas redes restringe-se exclusivamente à camada de saída. Como as coordenadas solares foram submetidas a um processo de discretização, o problema de regressão foi convertido em classificação categórica. Dessa forma, a rede voltada ao zênite possui uma camada final contendo 16 neurônios correspondentes a cada classe, enquanto a rede do azimuth é finalizada com 32 neurônios. Essa configuração final com a função de ativação *Softmax* permite que a rede calcule a probabilidade de o sol estar localizado em um dos intervalos angulares específicos mapeados. As Figuras 12 e 13 detalham os hiperparâmetros e as dimensões de cada camada dessas arquiteturas.

A terceira rede neural compartilha a mesma base convolucional dos modelos anteriores, porém distingue-se pelo seu objetivo central: a estimativa dos parâmetros de iluminação da cena com base no modelo analítico de Lalonde-Matthews. A principal alteração arquitetural concentra-se na camada de saída. Por se tratar de um modelo de regressão multivariada, a última camada não aplica uma função de ativação não linear, fornecendo como saída cinco valores numéricos contínuos, em que cada unidade representa um parâmetro físico específico do modelo.

De acordo com o trabalho de Lalonde e Matthews (LALONDE; EFROS; NA-

Figura 12 – Arquitetura da rede neural que encontra o azimuth do sol

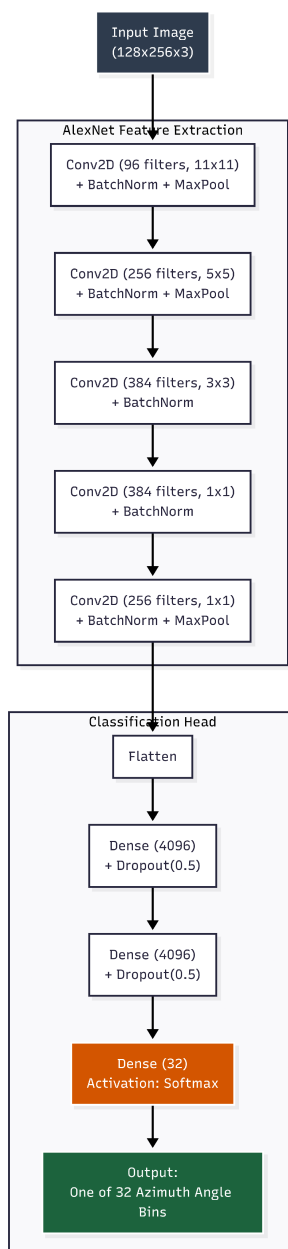
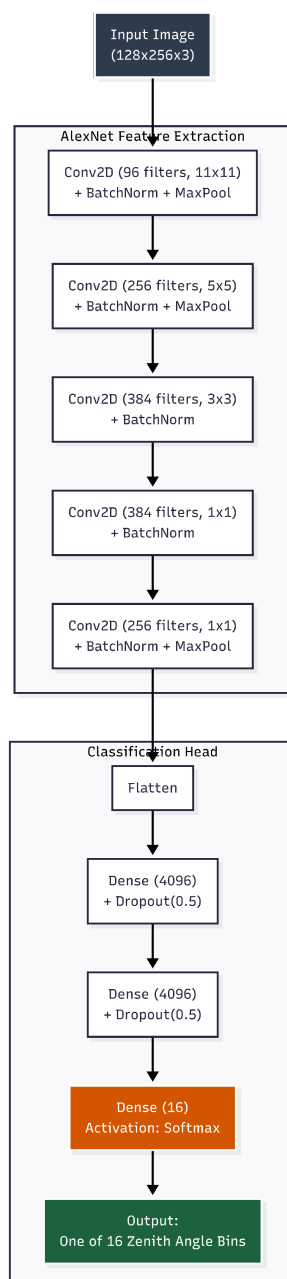


Figura 13 – Arquitetura da rede neural que encontra o zênite do sol



RASIMHAN, 2012), os cinco parâmetros que compõem este modelo de iluminação da abóbada celeste são definidos da seguinte forma:

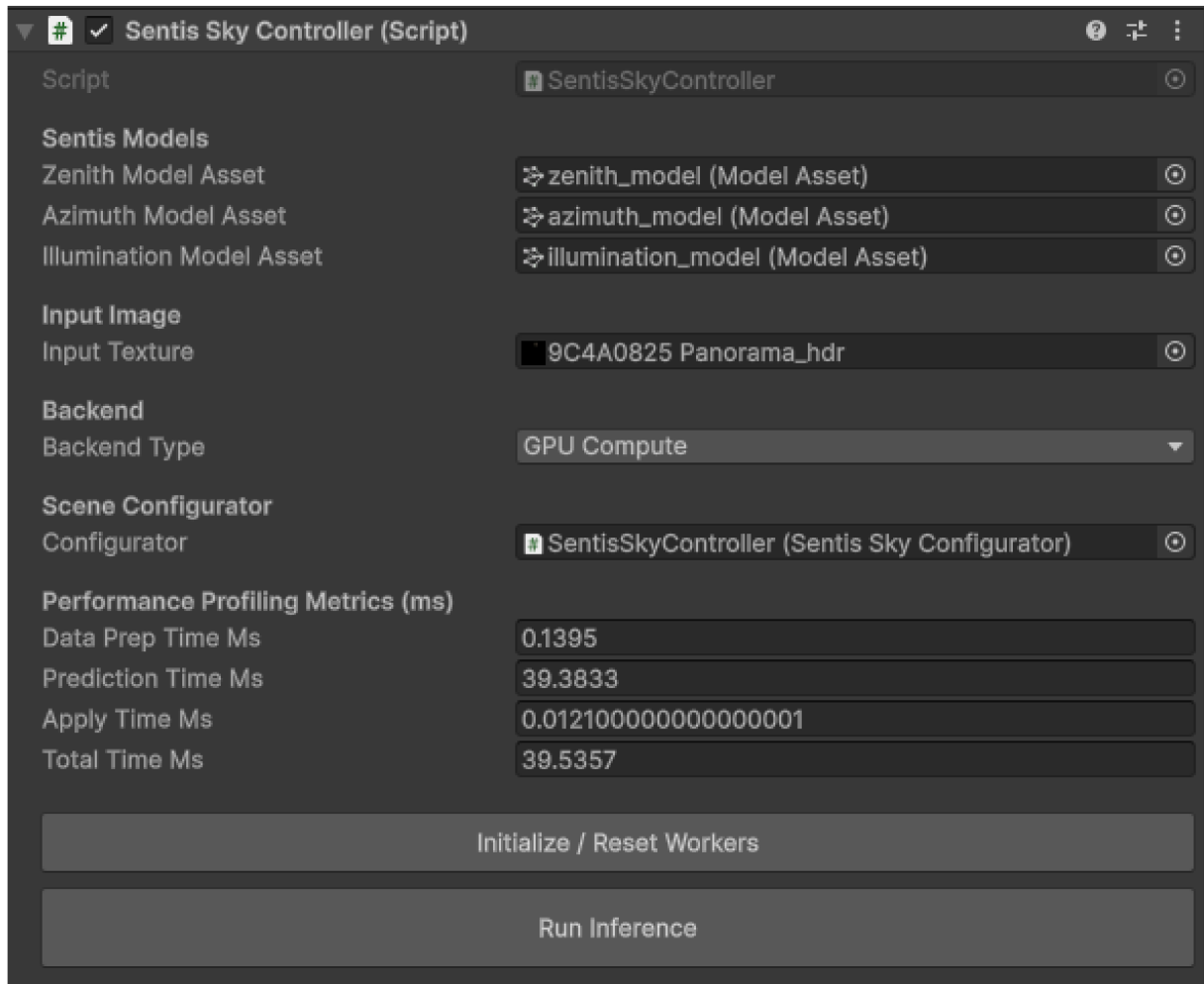
- **Beta** (β): representa o tamanho visual do disco solar.
- **Kappa** (κ): define o gradiente de decaimento do brilho ao redor do sol, correspondendo ao efeito visual de halo.
- **Turbidez** (t): indica a claridade ou nebulosidade do céu, sendo diretamente influenciada pela quantidade de partículas em suspensão ou poeira na atmosfera.
- W_{sun} : corresponde à intensidade absoluta da emissão de luz proveniente diretamente da fonte solar.
- W_{sky} : representa a intensidade da luz difusa emitida por todo o domo celeste.

A rotina de treinamento das redes foi estruturada empregando técnicas avançadas de gerenciamento de dados e otimização dinâmica. Para viabilizar o processamento do alto volume de imagens sem exceder a capacidade de memória do sistema, utilizou-se a API *tf.data* do TensorFlow, permitindo o carregamento e a decodificação sob demanda com paralelização de *hardware*. Adicionalmente, implementou-se o uso de *Callbacks* durante o ciclo de épocas, incorporando o salvamento condicional do modelo, a parada antecipada para mitigação de *overfitting* e a redução dinâmica da taxa de aprendizado visando uma sintonia fina nas etapas finais da convergência.

Devido à natureza distinta das saídas de cada modelo, fez-se necessária a adoção de diferentes funções de perda para a etapa de treinamento. As redes dedicadas à estimativa do zênite e do azimute, por se tratarem de modelos de classificação categórica resultantes da discretização dos ângulos, foram compiladas utilizando a Entropia Cruzada Categórica. Essa função avalia a divergência entre a distribuição de probabilidade prevista pela rede e as classes reais. Em contrapartida, a rede destinada à extração dos parâmetros de iluminação foi modelada como um problema de regressão multivariada. Para esta, adotou-se o Erro Quadrático Médio, garantindo que grandes desvios na predição das grandezas físicas contínuas (como a intensidade absoluta da luz solar) fossem severamente penalizados durante o ajuste dos pesos da rede.

A aplicação prática dos modelos de redes neurais desenvolvidos foi realizada dentro do motor gráfico Unity 6.3, utilizando a biblioteca Unity Sentis 2.5. Este *framework* permite a importação de modelos no formato ONNX (*Open Neural Network Exchange*) e a execução nativa da inferência, tirando proveito da aceleração de *hardware* disponível na máquina do usuário.

Figura 14 – Captura mostrando os parâmetros do componente Sentis Sky Controller



A arquitetura do sistema foi estruturada em três componentes principais para garantir uma separação limpa entre a execução da rede neural, a aplicação visual na cena e as ferramentas de interface do editor.

O núcleo de execução é o componente *SentisSkyController*. Este script é responsável por gerenciar o ciclo de vida dos *Workers* do Sentis e preparar os dados. Quando uma imagem panorâmica é fornecida, o controlador converte a textura de entrada em um tensor multidimensional no formato NHWC (*Batch, Height, Width, Channels*), com dimensões de $1 \times 128 \times 256 \times 3$. Esse tensor é então paralelizado e processado simultaneamente pelas três redes neurais (Zênite, Azimute e Iluminação).

Após a inferência, os tensores de saída são transferidos da GPU de volta para a memória da CPU. Para as redes espaciais, aplica-se a operação de Argmax para decodificar o vetor de probabilidades e extrair a classe com maior índice de confiança, convertendo-a novamente para os ângulos em radianos. A Figura 14 mostra como é exibido esse componente na Unity.

A representação fotorrealista de um ambiente diurno na Unity exige a atuação

conjunta de dois elementos fundamentais: a *Directional Light* e um material de *Skybox* procedural. O script *SentisSkyConfigurator* atua como a ponte entre os resultados numéricos da rede neural e esses componentes visuais.

A *Directional Light* funciona como a fonte de emissão fotométrica primária, sendo a principal responsável por iluminar a geometria da cena e projetar sombras. Como as redes espaciais predizem a posição do sol em coordenadas esféricas globais, o configurador realiza a conversão matemática para o sistema de coordenadas cartesianas da Unity (em que o eixo Y representa a altura, quanto maior o valor, mais para cima):

$$x = \sin(\phi) \cdot \sin(\theta)$$

$$y = \cos(\phi)$$

$$z = \sin(\phi) \cdot \cos(\theta)$$

Após o cálculo do vetor de direção, a rotação da luz direcional é ajustada utilizando a função *Quaternion.LookRotation*, orientando os raios luminosos exatamente para a posição estimada pela rede. A intensidade dessa luz é diretamente modulada pelo parâmetro W_{sun} inferido pelo modelo de iluminação.

Enquanto a luz direcional afeta os objetos 3D, o *SkyMaterial* (Material Procedural de Céu) é responsável pela representação visual da abóbada celeste, do gradiente atmosférico e do próprio disco solar. Os parâmetros do modelo de Lalonde-Matthews foram mapeados para as propriedades nativas deste material de forma direta:

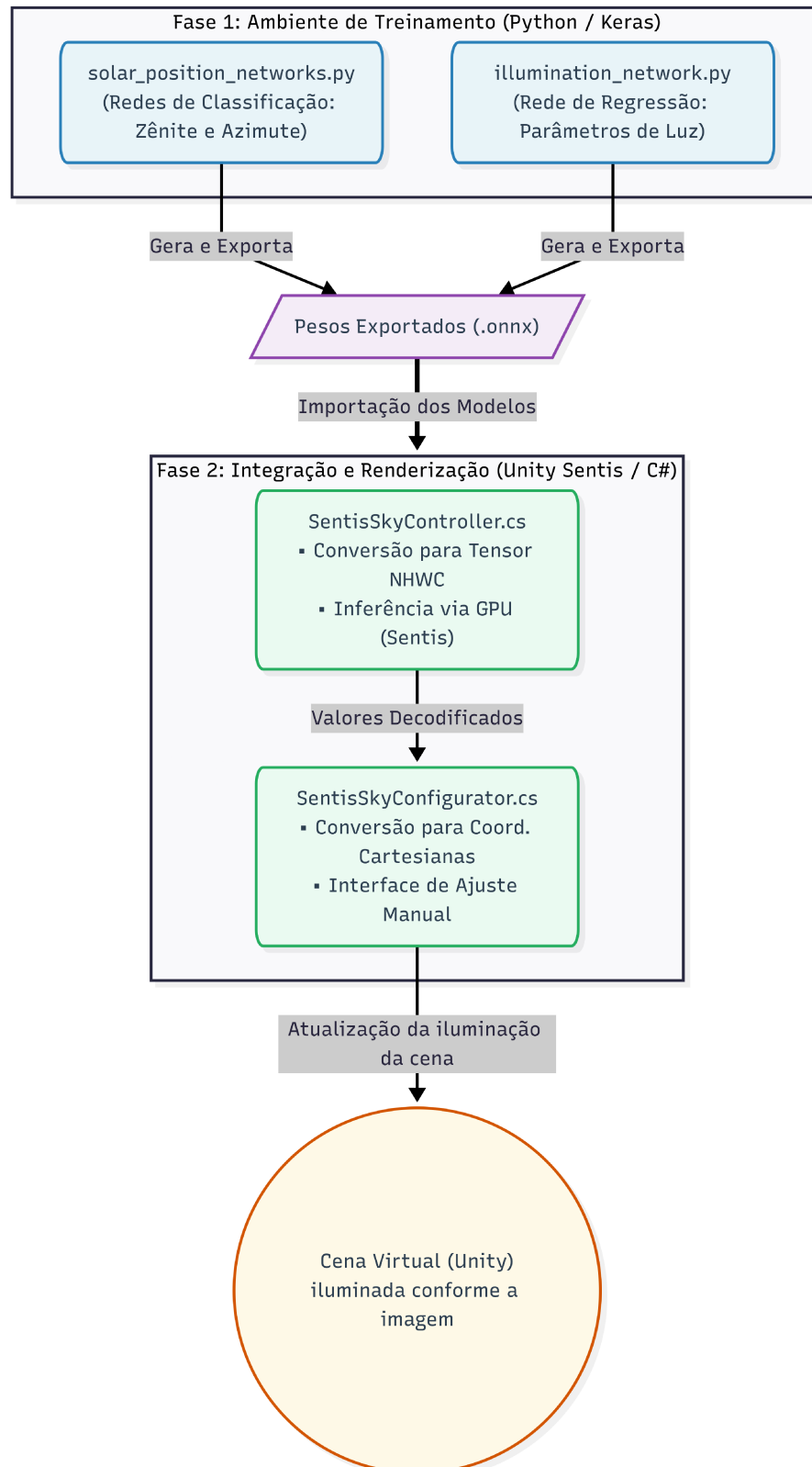
- Tamanho do Sol (β): Mapeado para a propriedade `_SunSize`, controlando a escala visual do disco solar no céu.
- Halo e Decaimento (κ): Inversamente mapeado para a convergência (`_SunSizeConvergence`), ajustando o espalhamento do brilho ao redor do sol.
- Turbidez Atmosférica (t): Mapeada tanto para a espessura da atmosfera (`_AtmosphereThickness`) no material do céu, quanto para a atenuação volumétrica da neblina global (`RenderSettings.fog`), simulando a degradação da visibilidade causada por partículas e poeira.
- Intensidade Global (W_{sky}): Mapeada para a exposição do material (`_Exposure`), controlando a claridade geral da abóbada celeste independente do sol.

Para facilitar o fluxo de trabalho e o ajuste fino (*fine-tuning*) da cena, o sistema foi encapsulado com a diretiva `[ExecuteAlways]`. Isso permite que os parâmetros da rede neural, uma vez extraídos, possam ser visualizados e ajustados manualmente através de controles deslizantes (*sliders*) no inspetor, atualizando a iluminação da cena em tempo

real sem a necessidade de iniciar o modo de jogo. Adicionalmente, implementou-se um script de editor (SentisSkyControllerEditor) que automatiza a configuração inicial da cena, vinculando automaticamente a Directional Light e o Skybox ativos ao controlador da inteligência artificial.

A arquitetura de código desse projeto pode ser visualizada na Figura 15.

Figura 15 – Diagrama que demonstra a arquitetura da fase de treinamento da rede neural e da implementação na unity



5 Resultados

A validação da metodologia proposta foi conduzida por meio da avaliação individual das três redes neurais convolucionais desenvolvidas. Para garantir a confiabilidade das métricas e atestar a capacidade de generalização dos modelos, os testes foram executados exclusivamente sobre conjuntos de dados de validação, compostos por imagens que não foram apresentadas às redes durante a fase de treinamento.

5.1 Avaliação das Redes de Posição Solar (Zênite e Azimute)

A eficácia das duas redes dedicadas à estimativa da localização espacial da fonte luminosa foi mensurada utilizando um conjunto de testes contendo 672 recortes panorâmicos isolados. Para avaliar o desempenho arquitetural em um problema de classificação em que grandezas contínuas (ângulos) foram discretizadas em intervalos (16 classes para o zênite e 32 para o azimute), adotaram-se três métricas: Acurácia Exata, Acurácia com Tolerância (± 1 classe) e Erro Médio.

Em tarefas de predição espacial discretizada, a métrica de Acurácia Exata apresenta severas limitações analíticas, uma vez que penaliza de forma binária predições que divergem marginalmente da fronteira da classe real. Sob a ótica da renderização fotorrealista em motores gráficos como a Unity, o indicador de maior relevância é o Erro Médio. Um Erro Médio baixo garante que os desvios da rede neural sejam restritos a intervalos angulares adjacentes, o que se traduz em uma variação visual de iluminação e projeção de sombras praticamente imperceptível. Os resultados consolidados estão expostos na Tabela 1.

Tabela 1 – Resultados da avaliação das redes neurais de classificação espacial do sol.

Rede Neural	Erro Médio (Classes)	Acurácia Exata	Acurácia (± 1 classe)
Zênite (16 intervalos)	1,30	25,7%	59,7%
Azimute (32 circular)	5,48	11,6%	26,3%

A rede preditora do ângulo zênite demonstrou um desempenho altamente robusto, registrando um Erro Médio de apenas 1,30 classes de distância. Ao se aplicar a tolerância de um intervalo adjacente, a acurácia do modelo atinge 59,7%, indicando que, na grande maioria das inferências, a elevação da fonte luminosa calculada pela inteligência artificial está perfeitamente alinhada ou marginalmente próxima à realidade física do cenário original.

Em contrapartida, a estimação do ângulo azimuth revelou-se um desafio computacional de maior complexidade, resultando em um Erro Médio de 5,48 classes e uma acurácia com tolerância de 26,3%. Essa assimetria de desempenho em relação ao zênite é teoricamente justificada pela natureza da amostragem visual. O azimuth determina a direção horizontal; contudo, em recortes em que a câmera possui alta elevação, o horizonte geográfico e os referenciais de solo não estão visíveis. A ausência de marcos visuais e de projeções de sombras no terreno obriga a rede a inferir a cardinalidade unicamente por gradientes de difusão nas nuvens, gerando ambiguidades inerentes ao problema.

5.2 Avaliação da Rede de Parâmetros de Iluminação

Para atestar a precisão do modelo de regressão responsável pela extração dos parâmetros físicos baseados no modelo de Lalonde-Matthews, utilizou-se uma amostra de 735 imagens de validação. A métrica estatística adotada foi o Erro Absoluto Médio (MAE - *Mean Absolute Error*), calculado tanto na escala normalizada da rede (valores de 0 a 1) quanto na conversão para as grandezas do mundo real.

De forma global, a arquitetura obteve um Erro Absoluto Médio normalizado de 0,1387 (13,87%). Os resultados para cada variável física estão dispostos na Tabela 2.

Tabela 2 – Erro Absoluto Médio (MAE) para os parâmetros de iluminação de Lalonde-Matthews.

Parâmetro Físico	Erro Médio (Normalizado)	Margem de Erro (Mundo Real)
Beta (β - Tamanho)	0,1655 (16,5%)	$\pm 5,5265$ unidades
Kappa (κ - Halo)	0,1431 (14,3%)	$\pm 0,0588$ unidades
Turbidez (t)	0,1357 (13,6%)	$\pm 1,5240$ unidades
W_{sun} (Intensidade Sol)	0,1375 (13,8%)	$\pm 0,4228$ unidades
W_{sky} (Intensidade Céu)	0,1117 (11,2%)	$\pm 0,0696$ unidades

A análise dos componentes demonstra que o modelo possui uma excelente capacidade de interpretação da iluminação difusa. O parâmetro referente à intensidade global da abóbada celeste (W_{sky}) apresentou o menor índice de erro, 11,2%, seguido pelas estimativas climáticas de Turbidez e intensidade solar (W_{sun}), que se mantiveram estatisticamente próximas, na faixa de 13,6% e 13,8%.

O maior grau de incerteza da rede manifestou-se na regressão de Beta com 16,5% de erro, parâmetro que governa o tamanho visual do disco solar. Computacionalmente, essa oscilação é esperada devido ao fenômeno de *blooming* em imagens de Grande Alcance Dinâmico (HDR), em que a saturação dos pixels oculta a borda real do sol, dificultando a predição cirúrgica de sua escala.

Em suma, os dados indicam que as arquiteturas propostas conseguem estabelecer com estabilidade a atmosfera da cena (cor do céu, neblina e halo) e a direção primária da

luz, consolidando-se como um recurso válido para a configuração procedimental de céus dinâmicos em simulações virtuais interativas.

Para avaliar qualitativamente o desempenho do sistema integrado, procedeu-se a uma comparação visual direta entre dois elementos: a cena virtual renderizada no motor gráfico Unity utilizando os parâmetros reais do *Ground Truth* e a mesma cena renderizada a partir das previsões numéricas da rede neural.

A simulação no ambiente virtual foi estabelecida através de uma *Directional Light* e de um material procedural de *Skybox*, parametrizados segundo o modelo de iluminação de Lalonde-Matthews. A Figura 16 ilustra a imagem panorâmica segmentada utilizada como entrada para a inferência neste experimento.

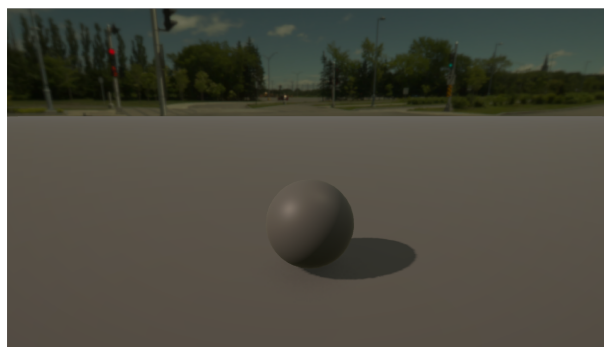
Figura 16 – Recorte utilizado para avaliar o modelo.



A imagem panorâmica original pertence ao banco de dados Laval Outdoor HDR Data-set ([LALONDE, 2019](#)).

Para estabelecer um referencial comparativo, a Figura 17 apresenta a renderização de uma cena virtual, composta por uma esfera central, iluminada estritamente pelos parâmetros reais (*Ground Truth*) vinculados à imagem original.

Figura 17 – Renderização da cena utilizando os parâmetros de iluminação reais (*Ground Truth*).

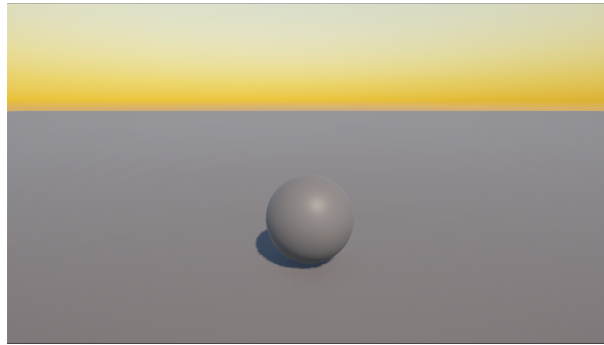


Em contrapartida, a Figura 18 exibe a mesma cena iluminada exclusivamente pelos parâmetros estimados pela rede neural. Ao confrontar as duas renderizações, nota-se uma divergência visual quanto à direcionalidade da luz. Na configuração *Ground Truth*

(Figura 17), a sombra é projetada à direita do ponto de vista da câmera. Já no resultado inferido pelo modelo, a sombra apresenta-se com menor extensão e é projetada à esquerda. Esse comportamento evidencia um desvio na predição da posição espacial do sol, particularmente no eixo horizontal (azimute).

Contudo, observa-se que as intensidades da iluminação global e da luz direta assemelham-se de forma expressiva em ambos os cenários. Esse resultado demonstra que, apesar da imprecisão geométrica da fonte luminosa, a rede de regressão conseguiu extrair com sucesso a atmosfera luminosa da imagem de entrada. Dessa forma, o sistema fornece uma base de iluminação coerente e robusta, permitindo que o usuário, por meio de ajustes finos na ferramenta desenvolvida, corrija rapidamente a direção da sombra e alcance o fotorrealismo desejado para a cena.

Figura 18 – Renderização da cena utilizando os parâmetros de iluminação estimados pela rede neural.



6 Conclusão

Este trabalho propôs o desenvolvimento e a integração de um sistema baseado em inteligência artificial para a automatização da iluminação fotorrealista em ambientes virtuais. Através da utilização de uma única fotografia como entrada, o objetivo principal foi extrair os parâmetros físicos e espaciais da abóbada celeste e aplicá-los diretamente ao motor gráfico Unity, mitigando a necessidade de ajustes manuais e empíricos por parte de artistas técnicos e desenvolvedores.

Em comparação ao trabalho (BERWALDT, 2021), que serviu como base metodológica para esta pesquisa, o presente estudo destaca-se pela modernização de toda a infraestrutura tecnológica e pela evolução na integração com o motor gráfico. Enquanto o estudo anterior utilizou o pacote *Barracuda* para a execução das inferências do modelo *ONNX* dentro do ambiente virtual, este trabalho adotou a biblioteca *Unity Sentis*, sua sucessora oficial e otimizada. Aliado ao uso da versão Unity 6, o novo ecossistema proporcionou um suporte aprimorado aos tensores multidimensionais, garantindo maior estabilidade, paralelização eficiente e comunicação direta com o *hardware* (GPU). Essas atualizações demonstram a constante necessidade de adaptação do método frente aos rápidos avanços das ferramentas de desenvolvimento de jogos.

Para implementar o projeto, a metodologia foi estruturada no treinamento de três Redes Neurais Convolucionais (CNNs), baseadas na arquitetura AlexNet. Duas dessas redes foram modeladas como classificadoras para estimar as coordenadas esféricas do sol (zênite e azimute), enquanto a terceira operou como um modelo de regressão multivariada para extrair os componentes atmosféricos baseados no modelo de iluminação de Lalonde-Matthews.

Os resultados quantitativos demonstraram a viabilidade técnica da abordagem. A estimação da elevação solar (zênite) e das propriedades difusas da atmosfera (como a intensidade do céu e a turbidez) obteve métricas de erro consistentes e baixas. Constatou-se que a rede neural foi altamente capaz de interpretar a claridade global da imagem e de prever o espalhamento da luz com fidelidade aceitável. Por outro lado, a pesquisa também evidenciou os desafios intrínsecos à extração de parâmetros direcionais horizontais (azimute) em imagens com ausência de referenciais geográficos, em que a rede apresentou maior dificuldade devido a ambiguidades visuais no mapeamento das nuvens.

Na etapa de validação qualitativa, a integração do modelo computacional com o motor Unity, através do framework Sentis, provou-se bem-sucedida. O sistema de controle desenvolvido (*SentisSkyConfigurator*) conseguiu traduzir as saídas matemáticas das redes neurais em propriedades visuais diretas, modulando a *Directional Light* e o material pro-

cedural de *Skybox*. Apesar das variações observadas na direcionalidade exata das sombras e na intensidade absoluta do disco solar, a atmosfera geral da cena (coloração, névoa e densidade) reproduziu coerentemente o cenário original.

Conclui-se que, embora a intervenção humana ainda se faça necessária para ajustes finos em busca do fotorrealismo absoluto, a ferramenta desenvolvida atinge seu propósito ao automatizar a estruturação inicial da iluminação, acelerando significativamente o fluxo de trabalho no desenvolvimento de simulações interativas e jogos digitais.

Como desdobramentos naturais desta pesquisa, sugere-se para trabalhos futuros: Evolução Arquitetural: A substituição da base convolucional AlexNet por arquiteturas mais modernas e profundas, visando melhorar a extração de características espaciais complexas e aumentar a acurácia na predição do azimute.

Expansão do Dataset: O treinamento dos modelos com um banco de imagens mais diversificado, englobando diferentes condições climáticas extremas e ciclos noturnos, para expandir a aplicabilidade do modelo de Lalonde-Matthews a cenários de baixa luminosidade.

Referências

- BERWALDT, N. L. P. Parametrização da iluminação de uma cena virtual 3D baseado em imagem LDR. Universidade Federal de Santa Maria, 2021. Citado 5 vezes nas páginas 6, 14, 16, 20 e 33.
- BHATTAD, A.; SOOLE, J.; FORSYTH, D. A. Stylitgan: Image-based relighting via latent control. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2024. p. 4231–4240. Citado 3 vezes nas páginas 3, 14 e 15.
- CHEN, X.; WANG, K.; JIN, X. Single image based illumination estimation for lighting virtual object in real scene. In: IEEE. **2011 12th International Conference on Computer-Aided Design and Computer Graphics**. [S.l.], 2011. p. 450–455. Citado na página 14.
- CHOI, H.-H.; KANG, H.-S.; YUN, B.-J. Cnn-based illumination estimation with semantic information. **Applied Sciences**, MDPI, v. 10, n. 14, p. 4806, 2020. Citado na página 16.
- CHOLLET, F. e. o. **Keras: Deep Learning for humans**. 2015. Disponível em: <<https://keras.io/>>. Acesso em: 27 abr. 2026. Citado na página 12.
- HOLD-GEOFFROY, Y.; ATHAWALE, A.; LALONDE, J.-F. Deep sky modeling for single image outdoor lighting estimation. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2019. p. 6927–6935. Citado na página 14.
- IORNS, T.; RHEE, T. Real-time image based lighting for 360-degree panoramic video. In: **Image and Video Technology**. [S.l.]: Springer, 2015. p. 139–151. Citado 4 vezes nas páginas 3, 14, 15 e 16.
- KHAN, A.; SOHAIL, A.; ZAHOORA, U.; QURESHI, A. S. A survey of the recent architectures of deep convolutional neural networks. **Artificial intelligence review**, Springer, v. 53, n. 8, p. 5455–5516, 2020. Citado na página 8.
- LALONDE, J.-F.; EFROS, A. A.; NARASIMHAN, S. G. Estimating the natural illumination conditions from a single outdoor image. **International Journal of Computer Vision**, Springer, v. 98, p. 123–145, 2012. Citado 2 vezes nas páginas 14 e 24.
- LALONDE, J.-F.; MATTHEWS, I. Lighting estimation in outdoor image collections. In: **2014 2nd International Conference on 3D Vision**. [S.l.: s.n.], 2014. v. 1, p. 131–138. Citado na página 14.
- LALONDE, J.-F. e. o. **The Laval Outdoor HDR Dataset**. 2019. Disponível em: <<http://hdrdb.com/outdoor/>>. Acesso em: 27 abr. 2026. Citado 2 vezes nas páginas 18 e 31.

- MIRBAUER, M.; RITTIG, T.; ISER, T.; KŘIVÁNEK, J.; ŠIKUDOVÁ, E. Skygan: Realistic cloud imagery for image-based lighting. In: WILEY ONLINE LIBRARY. **Computer Graphics Forum**. [S.l.], 2024. v. 43, n. 1, p. e14990. Citado na página 14.
- NEWZOO, B. **Newzoo global games market report 2022**. [S.l.]: Newzoo Reino Unido, 2022. Citado na página 6.
- OpenCV. **OpenCV: Open Source Computer Vision Library**. 2026. Disponível em: <<https://opencv.org/>>. Acesso em: 27 abr. 2026. Citado na página 13.
- Poly Haven. **Poly Haven: The Public 3D Asset Library**. 2026. Disponível em: <<https://polyhaven.com/hdris>>. Acesso em: 27 abr. 2026. Citado na página 18.
- PUSZTAHÁZI, L. S.; EIGNER, G.; CSISZÁR, O. Parametric activation functions for neural networks: a tutorial survey. **IEEE Access**, IEEE, v. 12, p. 168626–168644, 2024. Citado 2 vezes nas páginas 8 e 9.
- Unity Technologies. **Introduction to render pipelines**. 2026. Disponível em: <<https://docs.unity3d.com/6000.3/Documentation/Manual/render-pipelines-overview.html>>. Acesso em: 27 abr. 2026. Citado 4 vezes nas páginas 3, 9, 10 e 11.
- _____. **Introduction to the camera view**. 2026. Disponível em: <<https://docs.unity3d.com/6000.5/Documentation/Manual/UnderstandingFrustum.html>>. Acesso em: 21 jul. 2026. Citado 3 vezes nas páginas 3, 11 e 12.
- _____. **Plataforma de desenvolvimento em tempo real 3D | Unity**. 2026. Disponível em: <<https://unity.com/pt>>. Acesso em: 27 abr. 2026. Citado na página 9.
- _____. **Render pipeline feature comparison**. 2026. Disponível em: <<https://docs.unity3d.com/2022.3/Documentation/Manual/render-pipelines-feature-comparison.html>>. Acesso em: 27 abr. 2026. Citado na página 10.
- _____. **Sentis overview**. 2026. Disponível em: <<https://docs.unity3d.com/Packages/com.unity.ai.inference@2.6/manual/index.html>>. Acesso em: 27 abr. 2026. Citado na página 13.
- _____. **Types of Light component**. 2026. Disponível em: <<https://docs.unity3d.com/Manual/Lighting.html>>. Acesso em: 27 abr. 2026. Citado 3 vezes nas páginas 3, 10 e 12.