

Universidade Federal de Santa Catarina  
Centro de Blumenau  
Departamento de Engenharia de  
Controle, Automação e Computação



João Gabriel de Souza Godoy

Classificação de imagens residenciais para pesquisa  
personalizada em sites e chatbots do ramo imobiliário

Blumenau

2021

**João Gabriel de Souza Godoy**

**Classificação de imagens residenciais para pesquisa personalizada em sites e chatbots do ramo imobiliário**

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Santa Catarina como parte dos requisitos necessários para a obtenção do Título de Engenheiro de Controle e Automação.  
Orientador: Prof. Dr. Mauri Ferrandin

Universidade Federal de Santa Catarina  
Centro de Blumenau  
Departamento de Engenharia de  
Controle, Automação e Computação

Blumenau  
2021

João Gabriel de Souza Godoy

# Classificação de imagens residenciais para pesquisa personalizada em sites e chatbots do ramo imobiliário

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação.

**Comissão Examinadora**

---

Prof. Dr. Mauri Ferrandin  
Universidade Federal de Santa Catarina  
Orientador

---

Prof. Dr. Maiquel de Brito  
Universidade Federal de Santa Catarina

---

Prof. Dr. Marcos Vinicius Matsuo  
Universidade Federal de Santa Catarina

Blumenau, 22 de março de 2022

Dedico este trabalho a todos aqueles que, de alguma forma,  
auxiliaram para a concretização desta etapa.

# Agradecimentos

Agradeço a minha família, aos meus pais, Evaldo e Marilene, meus irmãos, Stela e Bernardo e meus sobrinhos Arthur e Isabela, por sempre me incentivarem e me apoiarem durante todo esse percurso.

Agradeço a Nathalia, minha namorada, a qual sempre esteve ao meu lado nessa caminhada e me deu suporte em todos os momentos difíceis, acreditando em mim e fazendo meus dias mais leves e alegres.

Agradeço a Universidade Federal de Santa Catarina, aos professores do curso de Engenharia de Controle e Automação do campus Blumenau pelos diversos ensinamentos e conhecimentos compartilhados e em especial ao meu orientador Mauri Ferrandin por todo o auxílio e empenho durante esse tempo de desenvolvimento.

Agradeço ao Grupo Brognoli e ao Nia Tecnologia para Negócios por terem me aceitado para o estágio e por ajudarem no meu desenvolvimento pessoal e profissional, em especial Ricardo, Geovane e Gabriel com quais as diversas conversas que tivemos foram de grande valia para a realização deste trabalho.



# Resumo

O mercado de crédito imobiliário brasileiro é um setor muito importante para economia do país e que representa uma quantia significativa de seu PIB, além disso, o setor imobiliário é um mercado muito competitivo em que o atendimento e a experiência do cliente são de extrema importância para o sucesso do seu negócio. Este trabalho visa desenvolver um sistema de busca personalizada que proporciona uma melhor experiência para a procura de imóveis possibilitando ao cliente escolher uma imagem, de um conjunto predefinido de imagens, que melhor define o imóvel pretendido e colocar como argumento de entrada na sua busca via chatbot ou página web. Este trabalho apresenta desde a criação dos conjuntos de dados contendo imagens de fachadas de casas residenciais, seu tratamento e rotulação, até o estudo e execução de algoritmos de visão computacional para a extração de características de imagens e técnicas de aprendizado de máquina, em especial Redes Neurais Convolucionais, para classificação. Por fim, todos os dados obtidos pelos diversos testes realizados são expostos e analisados.

**Palavras-Chave:** 1. Mercado imobiliário. 2. Redes neurais convolucionais. 3. Visão computacional.

# Abstract

The Brazilian real estate credit market is a very important sector for the country's economy that represents a significant amount of its GDP, furthermore, the real estate sector is a very competitive market in which customer service and experience are extremely important for the business success. This work aims to develop a personalized search system that provides a better experience for the search of properties, allowing the customer to choose an image, from a predefined set of images, that best defines the desired property and place it as an input argument in their search via chatbot or web page. This paper will cover from the creation of datasets containing images of residential house facades, their treatment and labeling, to the study and execution of computer vision algorithms for the extraction of image features and machine learning techniques, in particular Convolutional Neural Networks, for classification. Finally, all data obtained by the several tests performed will be exposed and analyzed.

**Keywords:** 1. Real estate market. 2. Convolutional Neural Networks. 3. Computer vision.

# Lista de figuras

Figura 1 – Construção de um modelo de classificação. . . . .	18
Figura 2 – Modelo de neurônio matemático . . . . .	19
Figura 3 – Ilustração do modelo mínimo de uma rede MLP . . . . .	20
Figura 4 – Ilustração de Rede Neural Artificial, de 4 camadas. . . . .	21
Figura 5 – Arquitetura básica de uma Rede Neural Convolutacional. . . . .	22
Figura 6 – Representação esquemática do descritor SIFT. . . . .	23
Figura 7 – Diagrama para cálculo e extração do descritor GIST . . . . .	24
Figura 8 – Estrutura geral da abordagem apresentada utilizando duas redes LSTM	25
Figura 9 – Diagrama de interações entre as entidade do Sistema Proposto. . . . .	29
Figura 10 – Diálogo atual entre usuário e chatbot para pesquisa de imóveis, diálogo realizado com a Fabi, chatbot da Imobiliária Brognoli. . . . .	30
Figura 11 – Diálogo proposto entre usuário e chatbot para pesquisa de imóveis, diálogo realizado com a Fabi, chatbot da Imobiliária Brognoli. . . . .	30
Figura 12 – Exemplo de imagens racionadas. . . . .	33
Figura 13 – Exemplo de imagens com zoom aplicado. . . . .	33
Figura 14 – Exemplo de imagens deslocadas em largura. . . . .	34
Figura 15 – Exemplo de imagens deslocadas em altura. . . . .	34
Figura 16 – Imagem modelo para classificação de imóveis como de alvenaria. . . . .	35
Figura 17 – Imagem modelo para classificação de imóveis como de madeira. . . . .	35
Figura 18 – Imagem modelo para classificação de imóveis como de tijolo a vista. . . . .	36
Figura 19 – Exemplos de imagens de imóveis que possuem características de mais de uma classe. . . . .	36
Figura 20 – Diagrama esquemático do primeiro modelo proposto. . . . .	38
Figura 21 – Diagrama esquemático do segundo modelo proposto. . . . .	39
Figura 22 – Diagrama esquemático do modelo VGG16 . . . . .	40
Figura 23 – Diagrama esquemático do modelo VGG19 . . . . .	40
Figura 24 – Diagrama esquemático do modelo ResNet50 . . . . .	41
Figura 25 – Diagrama esquemático do modelo InceptionV3 . . . . .	41
Figura 26 – Diagrama esquemático do modelo InceptionResNetV2 . . . . .	42
Figura 27 – Cross-validation, exemplo com número de pastas igual a 10. . . . .	43
Figura 28 – Representação gráfica do resultado de teste post-hoc de Nemenyi. . . . .	53

# Lista de tabelas

Tabela 1 – Descrição dos parâmetros para o primeiro modelo de CNN proposto. . . . .	39
Tabela 2 – Descrição dos parâmetros para o segundo modelo de CNN proposto. . . . .	39
Tabela 3 – Parâmetro escolhidos para os Modelos 1 e 2. . . . .	44
Tabela 4 – Distribuição das imagens entre classes para o primeiro conjunto de dados. . . . .	45
Tabela 5 – Distribuição das imagens entre classes para o segundo conjunto de dados. . . . .	45
Tabela 6 – Distribuição das imagens entre classes para o terceiro conjunto de dados. . . . .	45
Tabela 7 – Distribuição das imagens entre classes para o quarto conjunto de dados. . . . .	45
Tabela 8 – Resultados obtidos para a primeira abordagem, Extração de Features mais utilização algoritmos tradicionais, referentes ao primeiro conjunto de dados. . . . .	46
Tabela 9 – Resultados obtidos para as arquiteturas de CNN propostas referentes ao primeiro conjunto de dados utilizando Data Augmentation. . . . .	47
Tabela 10 – Resultados obtidos para as arquiteturas de CNN propostas referentes ao primeiro conjunto de dados sem utilizar Data Augmentation. . . . .	47
Tabela 11 – Resultados obtidos para a primeira abordagem, Extração de Features mais utilização algoritmos tradicionais, referentes ao segundo conjunto de dados. . . . .	47
Tabela 12 – Resultados obtidos para as arquiteturas de CNN propostas referentes ao segundo conjunto de dados utilizando Data Augmentation. . . . .	48
Tabela 13 – Resultados obtidos para as arquiteturas de CNN propostas referentes ao segundo conjunto de dados sem utilizar Data Augmentation. . . . .	48
Tabela 14 – Resultados obtidos para a primeira abordagem, Extração de Features mais utilização algoritmos tradicionais, referentes ao terceiro conjunto de dados. . . . .	48

Tabela 15 – Resultados obtidos para as arquiteturas de CNN propostas referentes ao terceiro conjunto de dados utilizando Data Augmentation. . . . .	49
Tabela 16 – Resultados obtidos para as arquiteturas de CNN propostas referentes ao terceiro conjunto de dados sem utilizar Data Augmentation. . . . .	49
Tabela 17 – Resultados obtidos para a primeira abordagem, Extração de Features mais utilização algoritmos tradicionais, referentes ao quarto conjunto de dados. . . . .	49
Tabela 18 – Resultados obtidos para as arquiteturas de CNN propostas referentes ao quarto conjunto de dados utilizando Data Augmentation. . . . .	50
Tabela 19 – Resultados obtidos para as arquiteturas de CNN propostas referentes ao quarto conjunto de dados sem utilizar Data Augmentation. . . . .	50

# Lista de Siglas e Abreviaturas

UFSC	<i>Universidade Federal de Santa Catarina</i>
PIB	<i>Produto Interno Bruto</i>
GDP	<i>Gross Domestic Product</i>
SVM	<i>Support Vector Machine</i>
ANNs	<i>Artificial Neural Networks</i>
MLP	<i>Multilayer Perceptrons</i>
DL	<i>Deep Learning</i>
CNN	<i>Convolutional Neural Network</i>
LSTM	<i>Long short-term Memory</i>
RGB	<i>Red, Green e Blue</i>
CO	<i>Coefficiente de Overfitting</i>

# Sumário

1	INTRODUÇÃO . . . . .	14
1.1	Contextualização do problema . . . . .	14
1.2	Objetivo Geral . . . . .	15
1.3	Objetivos Específicos . . . . .	15
2	REFERENCIAL TECNOLÓGICO . . . . .	16
2.1	Aprendizado de Máquina . . . . .	16
2.1.1	Redes Neurais Artificiais . . . . .	19
2.1.2	Multilayer Perceptron . . . . .	20
2.1.3	Aprendizado Profundo . . . . .	20
2.1.4	Redes Neurais Convolucionais . . . . .	21
2.2	Visão Computacional . . . . .	22
2.2.1	Descritores de características . . . . .	23
2.2.1.1	Descritor SIFT . . . . .	23
2.2.1.2	Descritor GIST . . . . .	24
3	REVISÃO DA LITERATURA . . . . .	25
3.1	Real Estate Image Classification . . . . .	25
3.2	Visual Estimation of Building Condition with Patch-level Conv- Nets . . . . .	26
3.3	Image Classification of Real Estate Images with Transfer Le- arning . . . . .	27
4	METODOLOGIA . . . . .	28
4.1	Arquitetura do Sistema Proposto e Contexto de Aplicação . .	28
4.2	Coleta de dados . . . . .	31
4.2.1	Pré-processamento . . . . .	32
4.2.1.1	Ajuste de escala . . . . .	32
4.2.1.2	Ajuste de tamanho . . . . .	32
4.2.1.3	Data augmentation . . . . .	32
4.2.2	Manual Data Labelling . . . . .	34
4.3	Abordagens para solução do problema . . . . .	36
4.3.1	Extração de Features mais utilização de algoritmos tradicionais	37
4.3.2	CNN específica para o problema . . . . .	38
4.3.3	CNN usando transfer learning . . . . .	40
4.4	Resumo dos experimentos e algoritmos testados . . . . .	42

5	EXPERIMENTOS E RESULTADOS . . . . .	43
5.1	Descrição dos Experimentos, Conjuntos de Dados, Parâmetros e Métricas . . . . .	43
5.1.1	Cross-validation . . . . .	43
5.1.2	Parâmetros de aprendizado das redes . . . . .	44
5.1.3	Métrica para avaliação dos modelos de classificação . . . . .	44
5.1.4	Conjunto de dados . . . . .	45
5.2	Resultados experimentais . . . . .	46
5.2.1	Resultados obtidos para o Primeiro Conjunto de Dados . . . . .	46
5.2.2	Resultados obtidos para o Segundo Conjunto de Dados . . . . .	47
5.2.3	Resultados obtidos para o Terceiro Conjunto de Dados . . . . .	48
5.2.4	Resultados obtidos para o Quarto Conjunto de Dados . . . . .	49
5.2.5	Análise de Dados . . . . .	50
6	CONCLUSÕES . . . . .	54
	REFERÊNCIAS BIBLIOGRÁFICAS . . . . .	56

# 1 Introdução

Este capítulo tem por objetivo contextualizar a problemática abordada por este trabalho, introduzir os conteúdos tratados e definir os objetivos gerais e específicos esperados.

## 1.1 Contextualização do problema

Entre os anos de 2020 e 2021 a pandemia de COVID-19 afetou muitos setores do comércio que precisaram procurar soluções tecnológicas e se digitalizar para poder atender seus clientes e também aumentar seu alcance. Para o setor imobiliário isso não foi diferente. Neste contexto, a procura e a utilização de ferramentas de comunicação e busca digitais são de grande importância tendo em vista que o contato humano foi reduzido drasticamente.

O mercado de crédito imobiliário brasileiro é um setor muito importante para o país e que sofreu um grande crescimento entre os anos de 2003 a 2018 passando de 1,5% para quase 10% do produto interno bruto (PIB) brasileiro [1].

Em decorrência da pandemia global da COVID-19 o mercado sofreu grandes baixas, mas já demonstra sinais de melhora sendo que no estado de Santa Catarina, no ano de 2020, foi registrado aumento de 26,6% nas operações imobiliárias. Outro dado que demonstra essa melhora é que nas maiores cidades do estado, Joinville e Florianópolis, houve aumento significativo nos registros de operações de compra e venda no período de abril de 2020 a abril de 2021, sendo 106,3% para Joinville e 211,6% para Florianópolis [2].

O mercado imobiliário é um ramo muito competitivo, onde a agilidade e a melhor experiência do cliente são grandes diferenciais na hora da escolha do cliente. Em grandes cidades a gama de imóveis à disposição é muito grande e diversificada, tornando a escolha difícil e demorada quando os aspectos físicos desejados para o imóvel não são levados em conta no momento da pesquisa.

Durante o período de estágio obrigatório, realizado no Grupo Brognoli, o qual gerencia nove empresas sendo duas delas a Imobiliária Brognoli, que atua em toda Grande Florianópolis, e a Nia Tecnologia para Negócios que trabalha com a automatização de processos imobiliários e o desenvolvimento de chatbots no ramo, foi constatado que são feitas cerca de 800 pesquisas de imóveis mensais somente via chatbot, podendo ter um número ainda maior para aqueles realizados diretamente ao site. Essas buscas são pautadas em perguntas que não levam em consideração aspectos físicos do imóvel, podendo assim, prejudicar a experiência do cliente, tornar longa e cansativa a procura, além de não ser assertiva.

Atualmente, visão computacional e aprendizado de máquina já são utilizados no ramo imobiliário tanto para pesquisas e trabalhos acadêmicos quanto em softwares comerciais,

algumas de suas aplicações são: prever o valor de mercado de imóveis [3], realizar reconhecimento e etiquetagem de imagens de cômodos da casa [4], estimar a condição de um imóvel por imagens e calcular sua desvalorização [5], avaliação de qualidade do imóvel entre outros.

Este trabalho aborda desde a criação do banco de imagens dos imóveis, suas classificações e operações realizadas até o estudo e a aplicação de extração de características de imagens e técnicas de aprendizado de máquinas, em especial Redes Neurais Convolucionais, e a definição de um mecanismo de integração do sistema desenvolvido com a potencial solução apresentada.

O desenvolvimento deste trabalho visa desenvolver um sistema de busca de imóveis que possibilite uma melhor experiência para o cliente durante as interações com o chatbot ou na página web da imobiliária, uma imagem de um conjunto pré definido de imagens que melhor definem o estilo e as características gerais do imóvel que ele procura.

## 1.2 Objetivo Geral

Este trabalho tem como objetivo geral estudar e aplicar técnicas de aprendizado de máquina e visão computacional à classificação de imagens residenciais para que a partir de suas características possa ser utilizado como parâmetro de entrada para buscas em sites imobiliários.

## 1.3 Objetivos Específicos

Para atingir o objetivo geral proposta, o trabalho deve alcançar os seguintes objetivos específicos:

- Estudar técnicas de extração de características de imagens;
- Estudar técnicas de classificação de imagens residenciais;
- Desenvolver um conjunto de dados para teste;
- Aplicar técnicas de extração de características de imagem no conjunto de dados;
- Desenvolver e aplicar técnicas de classificação de imagens residenciais no conjunto de dados;
- Definir um mecanismo de integração da aplicação.

## 2 Referencial tecnológico

Este capítulo tem por objetivo expor uma revisão teórica sobre os principais temas e tecnologias referentes ao desenvolvimento deste trabalho. Nele são abordadas revisões teóricas sobre os temas de aprendizado de máquina e visão computacional.

### 2.1 Aprendizado de Máquina

Como definido em [6] aprendizado de máquina é a programação de computadores para otimizar um critério de desempenho usando dados de exemplo ou experiências anteriores. Tendo um modelo definido em alguns parâmetros, a aprendizagem ocorre pela otimização desses parâmetros do modelo utilizando dados de treinamento ou experiência passada à medida que o programa do computador é executado. Um modelo utilizado para realizar previsões é classificado como preditivo ou descritivo para obter conhecimento dos dados, ou ambos.

Em particular, o processo de aprendizagem pode ser definido como descrito em [7], sendo: é dito que um programa de computador aprende a partir da experiência  $E$ , no que diz respeito a uma tarefa pré-determinada  $T$  e seu desempenho  $P$ , se o desempenho da tarefa  $T$ , medido por  $P$ , melhora com a experiência  $E$ .

Por exemplo, um algoritmo que aprende a classificar imagens de casas em três classes diferentes pode melhorar seu desempenho medindo a precisão de acerto dos rótulos das imagens pela experiência adquirida nas diversas vezes que realiza a classificação. Este é o cenário considerado neste trabalho.

As abordagens do aprendizado de máquina são caracterizadas dependendo da natureza do retorno que seu mecanismo de aprendizado, podendo ser elas: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço, onde conforme [8] define-se.

- **Aprendizado supervisionado:** é uma abordagem do aprendizado de máquinas definido pelo uso de conjuntos de dados rotulados. Usando entradas e saídas rotuladas, o modelo pode medir sua precisão e aprender com o tempo. Entre os algoritmos mais populares adequados ao aprendizado supervisionado temos: regressão linear e logística, árvore de decisão, máquinas de vetores de suporte (SVM) e redes neurais.
- **Aprendizado não supervisionado:** é a utilização de algoritmos de aprendizado de máquina para analisar e agrupar conjuntos de dados não rotulados, assim, descobrindo padrões presentes no conjunto de dados. Entre os algoritmos mais populares ade-

quados ao aprendizado não supervisionado temos: agrupamento hierárquico, redes neurais, agrupamento K-means.

- Aprendizado por reforço: é a abordagem de aprendizado de máquina interessada em fazer o agente aprender a aperfeiçoar suas ações a fim de obter maior recompensa cumulativa possível dentro de um ambiente em particular.

O aprendizado de máquina supervisionado pode ser dividido em duas subcategorias: classificação e regressão. Na classificação o objetivo é aprender uma função  $F$  que mapeie cada conjunto de atributos  $X$  para um dos rótulos de classes  $T$  pré-determinados. Por outro lado, na regressão aprender uma função alvo  $F$  que mapeie cada conjunto de atributos  $X$  para uma variável de saída contínua  $T$ . Neste trabalho é tratado o aprendizado de máquina supervisionado por classificação.

Cada modelo gerado pelo algoritmo de aprendizado de classificação deve se adaptar ao conjunto de dados de entrada do sistema e prever com precisão os rótulos das classes, pré determinados para cada instância de entrada, para exemplos de dados que não foram exibidos anteriormente ao algoritmo. Desta maneira, algoritmos de aprendizado devem ter boa capacidade de generalização como seu principal objetivo. A Figura 1 demonstra de maneira geral o problema de classificação que é discutido neste trabalho.

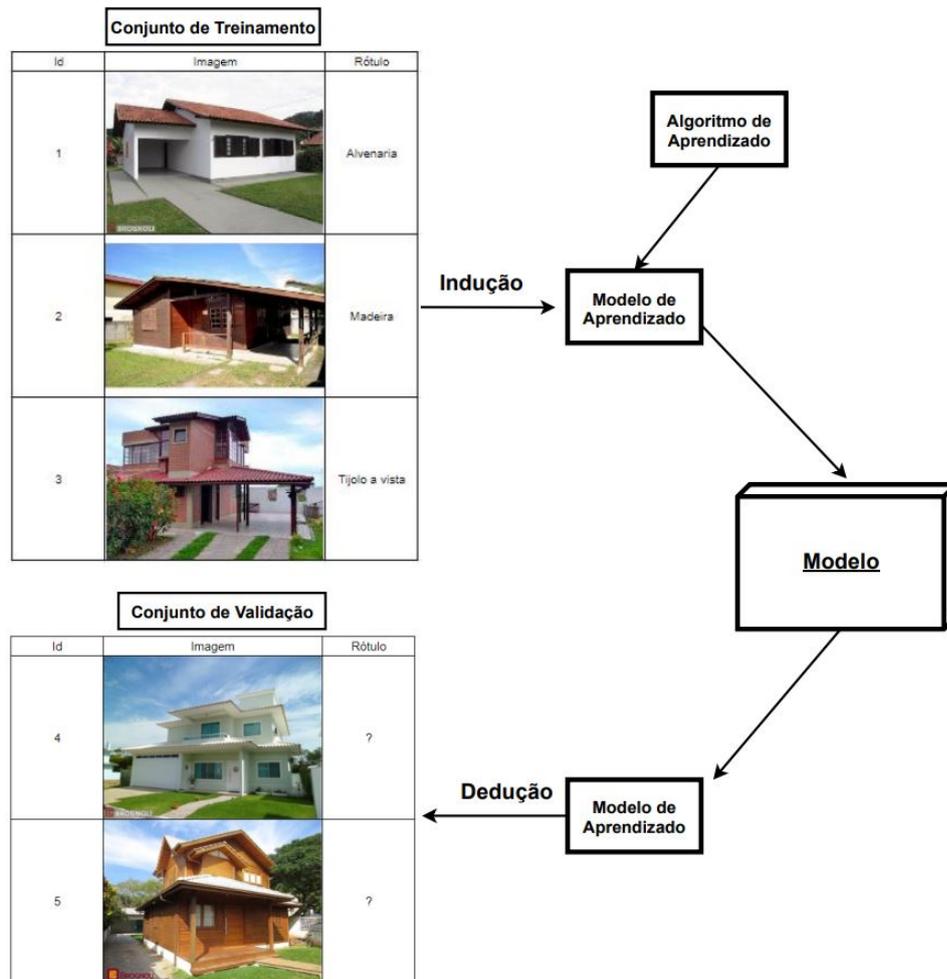


Figura 1 – Construção de um modelo de classificação, com base em [9].

Os problemas de classificação de dados ainda são divididos em: Plana, na qual o atributo classe pode assumir um dos valores pertencentes ao conjunto de classes, no caso do exemplo da Figura 1 Alvenaria, Madeira, Tijolo, e Hierárquica, na qual as classes estão pré dispostas dentro de uma estrutura hierárquica de classes, podendo assim pertencer a um conjunto de classes que respeitam a estrutura. Ambas ainda podem ser subdivididas em Monorrótulo, uma única classe é atribuída a instância, e Multirrótulo,  $n$  classes atribuídas à instância. E por último, a Classificação Plana Mono Rótulo pode ser subdividida em Binária, em que o atributo classe só pode assumir os valores binários {Sim, Não}, e Multiclasse na qual várias classes pré-determinadas podem estar associadas a uma instância. O problema de classificação aqui abordado é a Classificação Plana Monorrótulo Multiclasse.

Para resolução de problemas de classificação, no âmbito de aprendizado de máquina, são muitos os algoritmos tradicionais e estabelecidos na literatura como: KNN (*K-Nearest Neighbors*), SVM (Máquina de vetores de suporte), árvores de decisão, *Random Forests*, *Naive Bayes* e redes neurais os quais já foram amplamente discutidos e estudados e que são fundamentais para a base de conhecimento para classificação.

### 2.1.1 Redes Neurais Artificiais

As Redes Neurais Artificiais (ANNs) foram inspiradas pela observação de que os sistemas de aprendizado biológicos são construídos a partir de teias muito complexas de neurônios interconectados. Em analogia, as ANNs são construídas a partir de um conjunto densamente interconectado de unidades simples, neurônio, onde cada unidade recebe uma série de entradas e produz uma única saída [7]. Como definido em [10] o neurônio é uma unidade de processamento de informação o qual é fundamental para a operação de uma rede neural. A Figura 2 demonstra o modelo matemático não linear de um neurônio, descrito pelas equações Equação (2.1) e Equação (2.2), no qual são identificados e descritos 5 elementos básicos.

$$u_k = \sum_{j=1}^m w_{kj}x_j \quad (2.1)$$

$$y_k = \varphi(u_k + b_k) \quad (2.2)$$

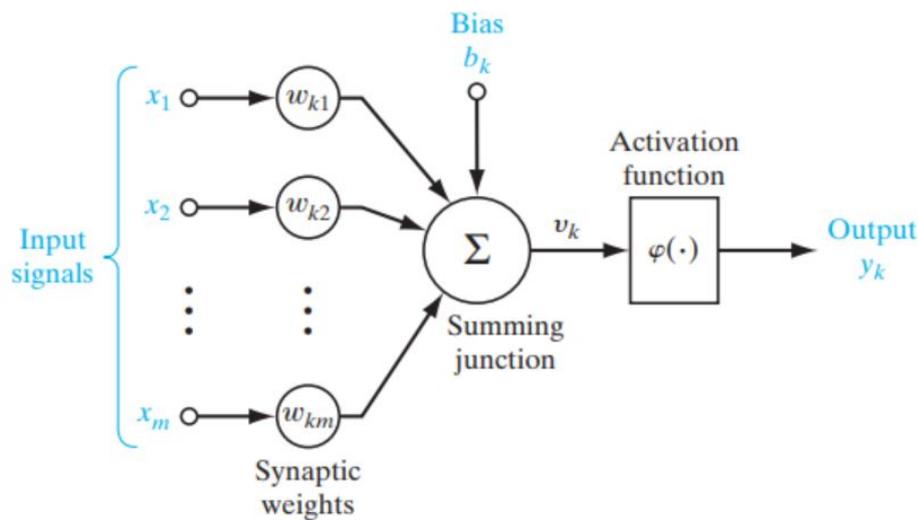


Figura 2 – Modelo de neurônio matemático, imagem retirada de [11].

- Sinais de entrada, que são um conjunto de sinapses, no qual uma sinapse  $j$  conectada ao neurônio  $k$  possui sinal de entrada  $x_j$  e peso sináptico  $w_{kj}$ .
- Um somador para realizar a combinação linear do conjunto de sinapses.
- O bias, ou viés, o qual tem a função de aumentar ou diminuir a entrada líquida da função de ativação.
- Função de ativação para limitar a amplitude do sinal de saída de um neurônio. Alguns exemplos de funções de ativação populares são: sigmoid, tangente hiperbólica, linear retificada (ReLU) entre outras.

- Sinal de saída, sinal resultante das operações realizadas com os sinais de entrada.

### 2.1.2 Multilayer Perceptron

As redes neurais compostas por múltiplos perceptrons, modelo de neurônio artificial, organizados em camadas são denominadas Multilayer Perceptrons (MLP). Uma MLP pode conter diversas camadas ocultas, porém sua construção mínima, Figura 3, é composta de 3 camadas sendo elas: uma camada de entrada, uma camada oculta e uma camada de saída. Cada nó presente na rede é um neurônio que possui uma função de ativação não linear, com exceção da camada de entrada.

Usualmente as redes MLP utilizam um algoritmo de *backpropagation* para realizar o cálculo da derivada da função custo, o qual quantifica a diferença entre o resultado esperado e o resultado obtido pelo modelo, em relação a cada parâmetro da rede. Enquanto outro algoritmo, como a descida do gradiente, é utilizado para realizar as alterações dos pesos dos parâmetros a partir dos resultados da função custo, desta maneira, efetuando o aprendizado da rede.

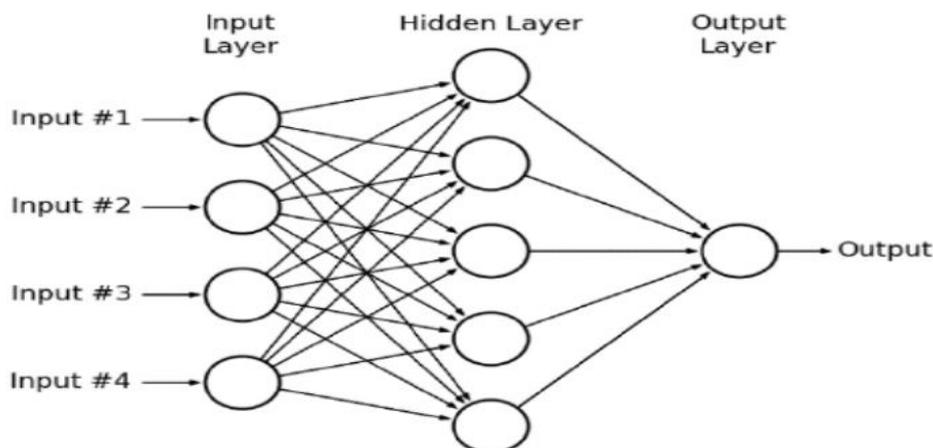


Figura 3 – Ilustração do modelo mínimo de uma rede MLP, retirado de [3].

### 2.1.3 Aprendizado Profundo

Aprendizado profundo (*Deep learning*, DL) é um subcampo do aprendizado de máquina que exclusivamente usa múltiplas camadas interconectadas de neurônios, como demonstra a Figura 4, para extrair padrões e recursos de dados brutos. Algoritmos de aprendizado profundo são especialmente utilizados para processamento de imagem, reconhecimento de voz e tradução automática [8].

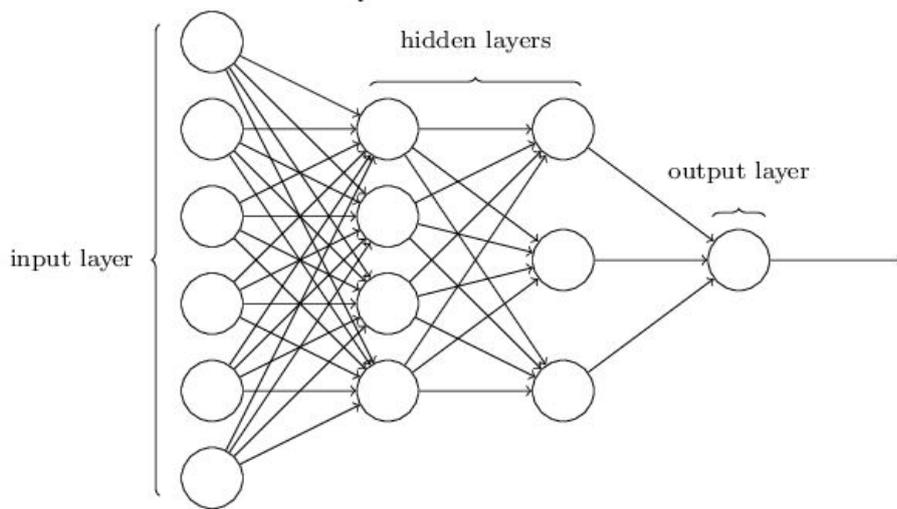


Figura 4 – Ilustração de Rede Neural Artificial, de 4 camadas.

### 2.1.4 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (*ConvNet / Convolutional Neural Network / CNN*) são um tipo especializado de redes neurais para o processamento de dados que possuem correlação espacial entre seus elementos vizinhos. Os exemplos incluem dados de série temporal como uma grade unidimensional obtendo amostras de tempos em intervalos regulares e imagens que podem ser considerados uma grade bidimensional de pixels.

As redes convolucionais são redes neurais que utilizam a operação de convolução no lugar de multiplicação geral de matriz em pelo menos uma de suas camadas [12]. A convolução é uma operação matemática em duas funções ( $x$  e  $w$ ) que produz uma terceira função ( $f * w$ ) a qual expressa como a forma de uma é modificada pela outra e é definida como a integral do produto das duas funções depois que uma é invertida e deslocada. A integral é avaliada para todos os valores de deslocamento, produzindo a função de convolução.

$$s(t) = \int x(a)w(t - a)da \quad (2.3)$$

A operação de convolução é tipicamente denotada utilizando o asterisco:

$$s(t) = (x * w)(t) \quad (2.4)$$

Na terminologia de redes convolucionais, o primeiro argumento da convolução, neste caso a função  $x$ , é usualmente referenciado como entrada, o segundo argumento, função  $w$ , como filtro e a saída como mapa de características.

A arquitetura de uma CNN consiste, usualmente, em diversas camadas de convolução e de agrupamento (*pooling*), que são utilizadas para para simplificar a complexidade dos dados da imagem, além disso, essas camadas realizam a extração de padrões complexos

observados no conjunto de dados. Depois dessas camadas de convolução e agrupamento a forma desses dados é remodelada para uma matriz unidimensional mediante o uso de uma camada de achatamento (*Flatten*). Após a camada de achatamento, esses dados são usados como entradas para um conjunto de camadas totalmente conectadas, a qual completa a tarefa classificação ou regressão [8].

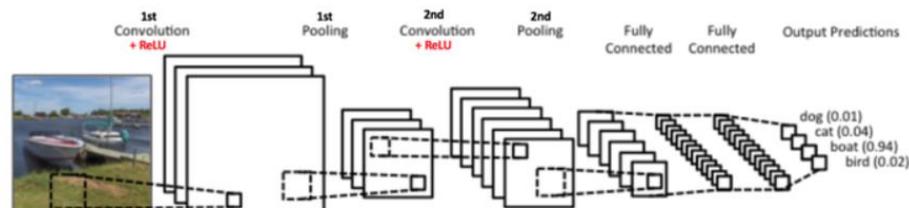


Figura 5 – Arquitetura básica de uma Rede Neural Convolutiva.

Em resumo, uma rede convolutiva apresenta as seguintes camadas:

- Camada de convolução: é onde é feita a extração de características de imagens. A função dessa camada é aplicar um filtro, menor que a imagem, para diminuir o seu tamanho sem perder a correlação entre os pixels da imagem.
- Camada de agrupamento: é utilizada para reduzir o tamanho espacial da imagem, assim, reduzindo o número de parâmetros da rede e conseqüentemente a complexidade computacional, além disso, auxilia em problemas de sobreajuste (*overfitting*).
- Camada de achatamento: é utilizada para remodelar os dados de matrizes bidimensionais ou tridimensionais para matrizes unidimensionais, assim, esses dados agora podem entrar para a camada totalmente conectada.
- Fully connected layer: é uma rede neural feedforward, no qual cada neurônio em uma camada é vinculado aos demais da próxima camada para determinar a relação e o efeito de cada parâmetro nos rótulos. A camada totalmente conectada tem a função de classificar as imagens no final da CNN.

## 2.2 Visão Computacional

Visão computacional é o campo da ciência da computação que trabalha na aquisição, processamento, análise e interpretação de imagens digitais e com base nessas interpretações fornece uma decisão apropriada para o problema proposto.

São três os principais tipos de características que possam ser extraídas de uma imagem por meio de algoritmos de visão computacional, sendo elas: características de regiões, características de linhas e características de pontos. As características de regiões descrevem

escalares ou vetores contendo informações sobre a posição, forma, orientação e regiões de interesse, as características de linhas apresentam um conjunto de parâmetros que definem a posição de uma linha de interesse na imagem e as características de pontos estabelecem coordenadas de pontos de interesse que são localmente distintos na imagem. Este trabalho é focado na utilização de características de ponto.

## 2.2.1 Descritores de características

Uma característica de ponto da imagem é um ponto de interesse definido pelas suas coordenadas em pixels  $[u,v]$  na imagem. Um descriptor é um vetor de  $N$  dimensões associado a cada ponto característico, o qual caracteriza ou identifica o ponto de interesse em questão. O descriptor deve ter as seguintes características:

- Repetibilidade: manifestada como a robustez e invariância à translação, rotação, diferentes escalas e mudanças de iluminação.
- Distinção: o descriptor deve permitir que duas características, mesmo que, sejam distinguidas.
- Compacto e eficiente: em relação à complexidade computacional.

### 2.2.1.1 Descritor SIFT

SIFT é uma técnica de processamento de imagem que permite a detecção e extração de descritores locais, razoavelmente invariáveis à mudanças de iluminação, ruído da imagem, rotação, escala e perspectiva. Tais descritores podem ser utilizados para operações de correspondência de um objeto em uma cena.

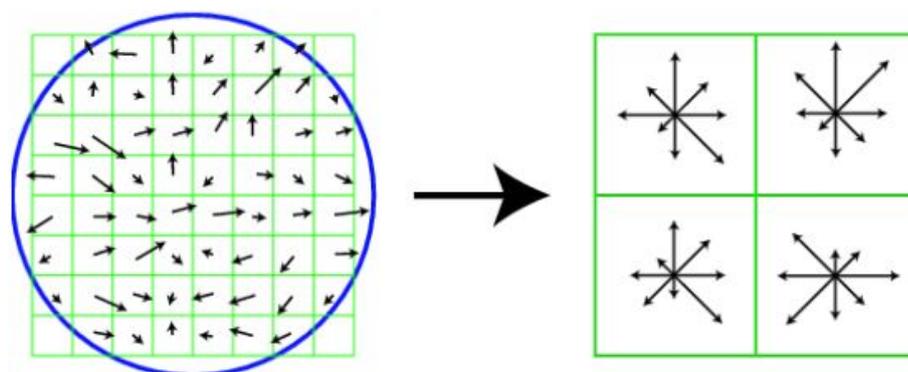


Figura 6 – Representação esquemática do descritor SIFT.

O procedimento para computar a extração de descritores SIFT começa por calcular o gradiente de cada pixel em uma janela  $16 \times 16$  pixels ao entorno de um ponto característico (feature) da imagem, as magnitudes dos gradientes são reduzidas utilizando a função de

queda Gaussiana para reduzir a influência dos gradientes distantes ao centro. Em cada quadrante 4x4 pixels é formado um histograma de orientação de gradiente pela adição do valor do gradiente ponderado a um dos oito compartimentos do histograma de orientação. Os 128 valores não negativos resultantes são então normalizados, a fim de evitar efeitos de contraste e ganho, formando o vetor descritor SIFT [13].

### 2.2.1.2 Descritor GIST

Em visão computacional os descritores GIST são a representação de uma imagem de baixa dimensão que contém informações suficientes para identificar a cena de uma imagem. O descritor global de uma imagem é construído combinando as amplitudes obtidas na saída dos filtros de Gabor a diferentes escalas e orientações.

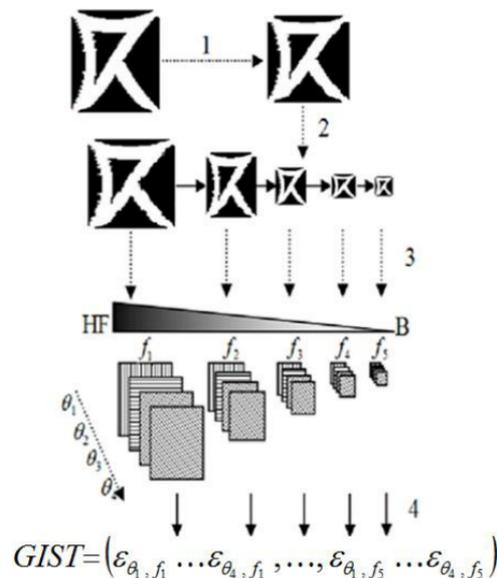


Figura 7 – Diagrama para cálculo e extração do descritor GIST, retirado de [14].

O processo de extração dos descritores GIST é o apresentado na Figura 7 que consiste em uma etapa de pré-processamento da imagem, em seguida a imagem é alterada em diferentes escalas e orientações e por último é efetuado o cálculo dos vetores de características para cada escala, contendo informações do gradiente (orientação e frequência). Esses vetores de características são combinados para formar um descritor de características global da imagem [14].

## 3 Revisão da Literatura

Este capítulo tem por objetivo expor o levantamento bibliográfico realizado na busca por artigos relacionados a utilização de redes neurais para classificação de imagens relacionados ao ramo imobiliário.

### 3.1 Real Estate Image Classification

Em [4] é apresentada uma arquitetura de modelo para classificação de imagens do ramo imobiliário. Em tal arquitetura, são executadas três tarefas de classificação: classificação do cômodo da casa, classificação do material de balcão e classificação do material do assoalho. Especificamente esses três classificadores apresentam a mesma arquitetura porém modelos distintos para tarefas distintas.

A arquitetura é baseada nos seguintes passos: realizar a melhora da qualidade da imagem utilizando a técnica de pré-processamento da equalização de histograma adaptativa limitada por contraste (CLAHE), em seguida a imagem é alimentada para duas redes long short term memory (LSTM) em paralelo, as quais têm a função de aprender a correlação entre pixels vizinhos, os recursos captados pelas redes são combinados a duas camadas totalmente conectadas, a última sendo um classificador softmax o qual é utilizado para prever os rótulos das imagens. O diagrama do processo é mostrado abaixo.

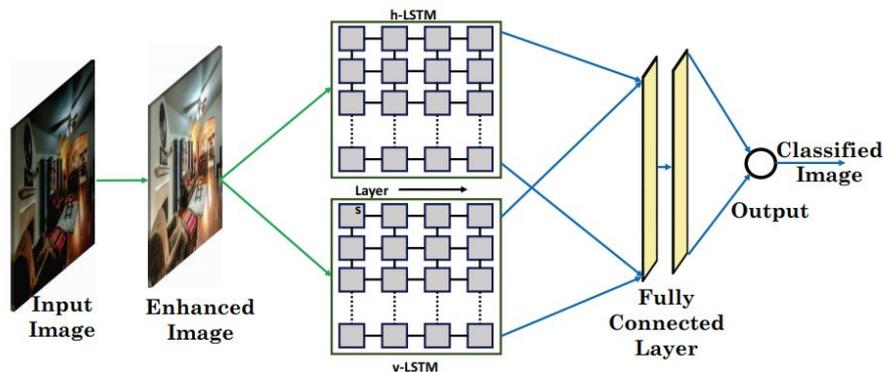


Figura 8 – Estrutura geral da abordagem apresentada utilizando duas redes LSTM, retirada de [4].

Os conjuntos de dados utilizados foram gerados a partir de imagens retiradas de listagens imobiliárias e de pesquisas na internet, sendo que para a classificação de cômodos foram utilizadas imagens de ambas as fontes e para a classificação de balcão e assoalho foram utilizados exclusivamente imagens retiradas da internet.

Os resultados experimentais de classificação obtidos pela arquitetura proposta superaram outros métodos de classificação já existentes, sendo alguns deles redes neurais convolucionais (CNN) como a Alexnet e VGG.

## 3.2 Visual Estimation of Building Condition with Patch-level ConvNets

Em [5] é abordado questões relacionadas à avaliação automática de condições de imóveis, baseando-se no pressuposto que o estado real do imóvel está representado na aparência visual de sua fachada. Em conjunto foi construído um modelo para prever o desconto atribuído a condição do imóvel.

Os autores observaram que diversos elementos arquitetônicos, além da fachada, como janelas, portas, telhado tem potencial para indicar o estado do imóvel, assim, percebendo que as informações relevantes para o problema possuem características locais. Com isto a abordagem do modelo se dá pela extração de características da imagem e com isso gera-se sub imagens de cada uma delas, sendo elas analisadas e classificadas separadamente.

O processo de classificação tem início a partir da extração de sub imagens da imagem principal utilizando amostragem regular para aumentar a heterogeneidade dos padrões visuais capturados. Após é utilizado o descritor SIFT, em cada uma das sub imagens, e é aplicado um algoritmo de agrupamento para selecionar quais irão continuar sendo utilizadas, com isso, removendo imagens com pouca informação. Em seguida é realizado o treinamento de uma rede CNN, modelo ResNet50 e utilizando os parâmetros pré-treinados no ImageNet, necessitando alterar apenas a camada de saída para se adequar a quantidade de rótulos preestabelecido. Por fim, para obter a estimativa do imóvel, são classificadas todas as sub imagens e qual rótulo obter a maioria dentre elas, permanece como rótulo a imagem principal.

As imagens podem ser classificadas em três categorias sendo elas: bom estado, condição normal e precisa de reparos. O conjunto de dados utilizado foi disponibilizado por uma imobiliária austríaca contendo cerca de 8000 imagens imobiliárias e metadados para os experimentos.

Por último é construído um modelo de regressão para prever o desconto atribuído a um imóvel, contendo os metadados do imóvel, a condição prevista e o ano de construção.

Os resultados dos experimentos realizados alcançaram a precisão de classificação de cerca de 65%, assim, mostrando que os padrões aprendidos pela rede são significativos e que o processamento baseado em sub imagens facilita a extração de padrões visuais. E em relação a predição de descontos são gerados resultados igualmente bons, na mesma faixa de acurácia da classificação, podendo assim, servir como intermediário para as estimativas de condição do imóvel auxiliando os avaliadores.

### 3.3 Image Classification of Real Estate Images with Transfer Learning

Em [15] são discutidos aspectos sobre filtros de pesquisa de imóveis em que além dos dados usuais, como número de quartos, preço e localização, existem outras características que podem ser decisivas na escolha do imóvel como varanda, forno a lenha e lareira. Assim, o trabalho busca classificar as imagens de anúncios de imóveis para prever se tal característica está presente e, assim, associar a característica ao imóvel sem ter a necessidade de estar descrita.

No trabalho são desenvolvidos três experimentos de classificação diferentes: modelo classificação binária para determinar se na imagem contém uma varanda ou não, modelo de classificação binária para determinar se na imagem contém lareira ou não e por fim modelo de classificação multiclasse para determinar qual cômodo do imóvel a imagem analisada representa.

A construção do conjunto de dados para realização dos experimentos foi realizada a partir da pesquisa das categorias desejadas e baixadas via Google Image Search. Foram selecionadas cerca de 400 imagens para cada rótulo de classificação.

Nos experimentos foram utilizados diferentes modelos de aprendizado de máquina para comparação da precisão alcançada em cada. Especificamente, as arquiteturas abordadas foram: Resnet18, Alexnet, VGG-11, Densenet-121 e Inception V3.

Os resultados obtidos nos experimentos foram muito significativos, alcançando mais de 95% em média para reconhecimento/classificação de varandas, chegando a 85% para arquitetura Densenet-121 na classificação de lareira e a quase 98% para classificação para as arquiteturas de VGG-11 e Densenet-121 na classificação de cômodos. Desses resultados pode-se concluir que as arquiteturas se adequam a situações diferentes.

# 4 Metodologia

Neste capítulo é apresentada a metodologia utilizada no desenvolvimento deste projeto, contemplando todos os passos percorridos para elaboração de um projeto de aprendizado de máquina partindo desde a coleta e tratamento dos dados até o treinamento e utilização de CNNs.

## 4.1 Arquitetura do Sistema Proposto e Contexto de Aplicação

O sistema proposto é composto por quatro entidades, a saber: o usuário, a interface para comunicação com o usuário, o classificador e o conjunto de dados. O processo se inicia no conjunto de dados, pois para tornar os dados contidos nele utilizáveis para a criação de um classificador é necessário que as imagens estejam de acordo com o que irá classificar, fachadas de casas residenciais, e devidamente rotuladas, assim que o conjunto de dados de treinamento está apto, ele se torna o conjunto de treinamento para a classificador. O próximo passo é criação de modelo de classificação e seu treinamento.

Em sequência o usuário acessa os meios de comunicação nos quais o sistema está aplicado, para este estudo consideram-se sites de imobiliárias e chatbots que fazem a consulta e pesquisa de imóveis, o qual irá retornar um conjunto de opções para que o usuário selecione a imagem que melhor represente o estilo de imóvel que ele procura. Isto feito, o site sinaliza ao classificador que houve uma seleção de imagem, assim, as imagens de fachadas de casas residenciais são requisitadas ao conjunto de dados que retorna as imagens, com estes dados o classificador é capaz de classificar as imagens e posteriormente selecionar as de mesma classe que a da escolha do usuário. Na sequência, o classificador retorna ao site/chatbot os imóveis selecionados que por sua vez exibe ao usuário na sua interface. O diagrama das interações acima descritas está na Figura 9.

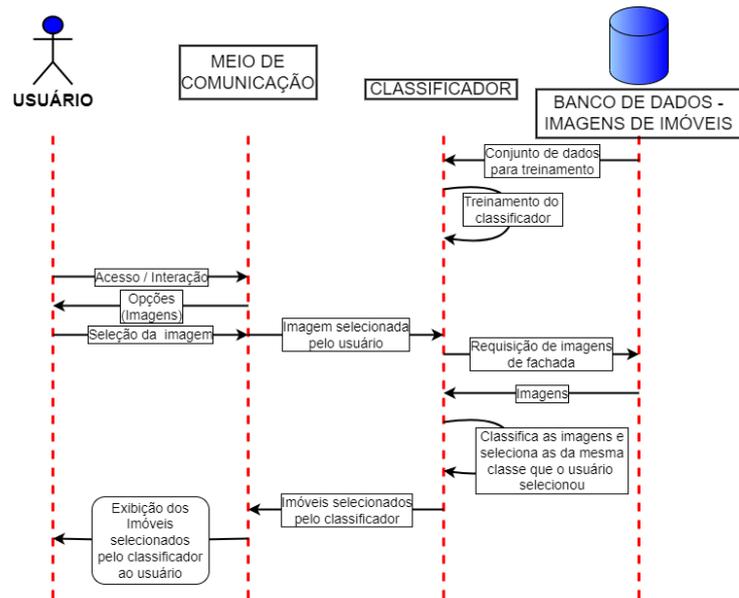


Figura 9 – Diagrama de interações entre as entidade do Sistema Proposto.

Como descrito no Capítulo 1, este trabalho tem como um de seus objetivos desenvolver um sistema que melhore a experiência do cliente na escolha de um imóvel com variáveis de entrada visuais. Desta maneira a Figura 10 mostra o panorama atual de pesquisas realizadas via chatbots e na Figura 11 é demonstrado como o fluxo de de conversação será utilizando a solução proposta neste trabalho.

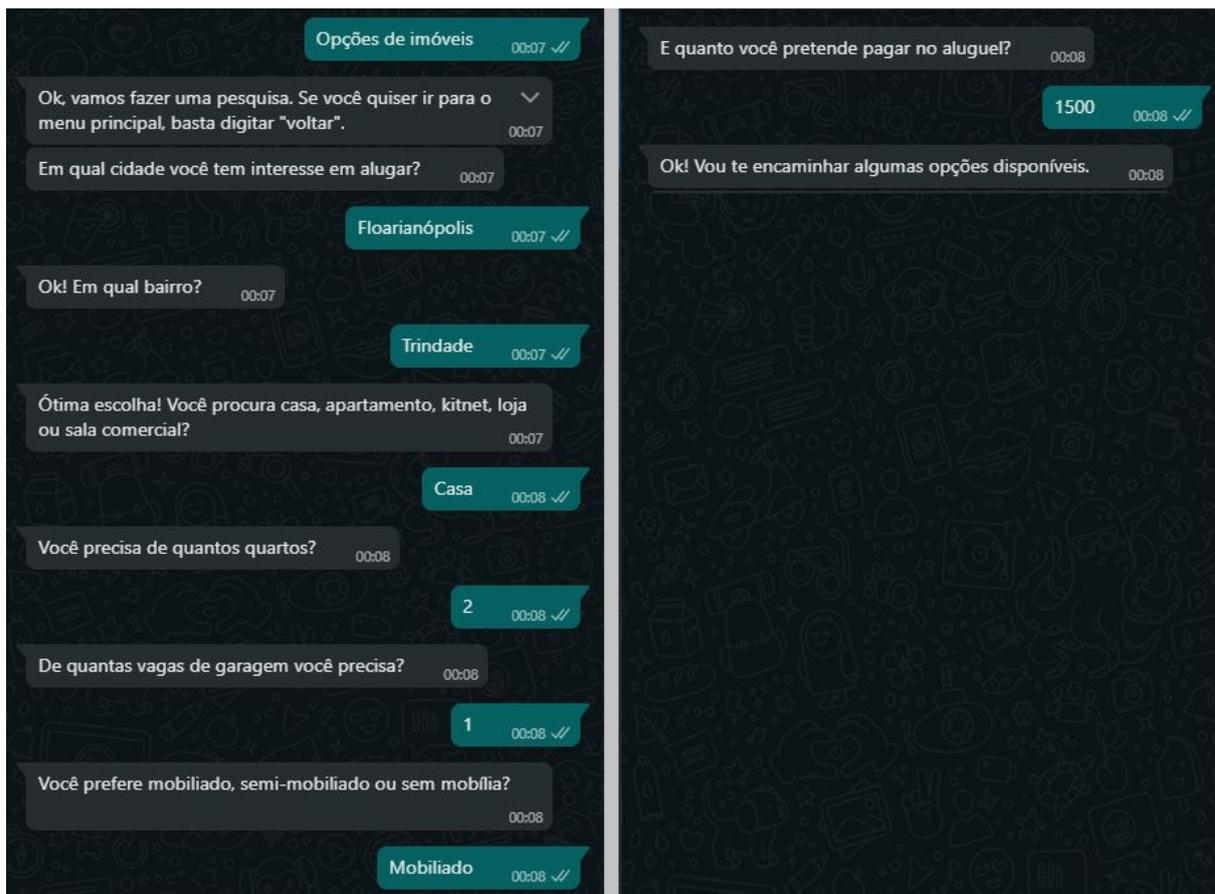


Figura 10 – Diálogo atual entre usuário e chatbot para pesquisa de imóveis, diálogo realizado com a Fabi, chatbot da Imobiliária Brognoli.

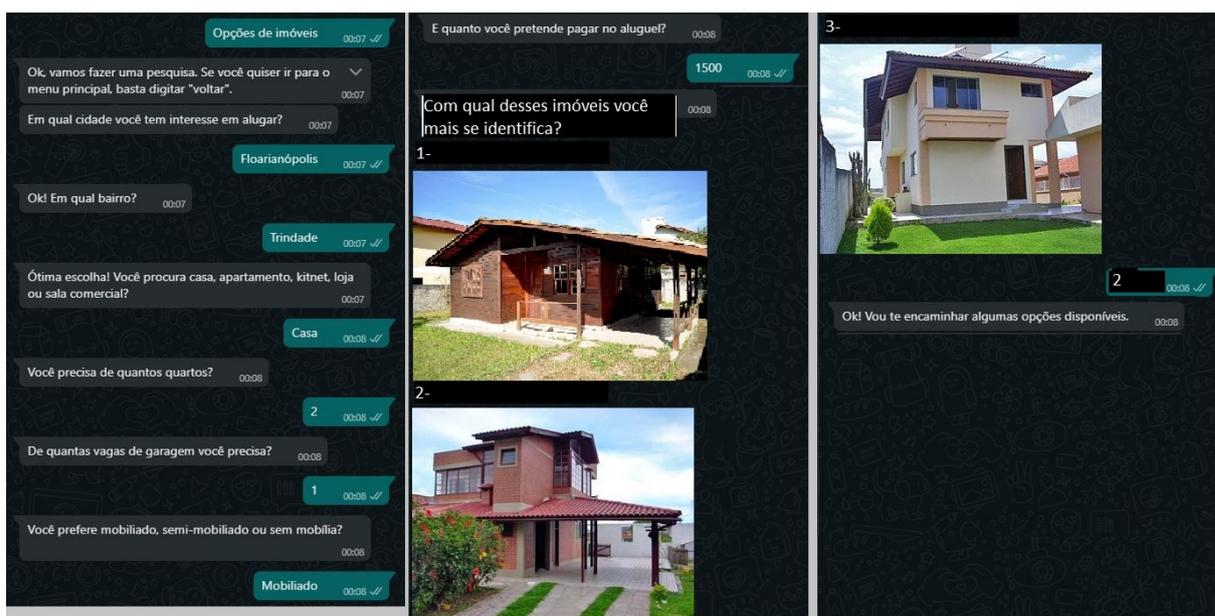


Figura 11 – Diálogo proposto entre usuário e chatbot para pesquisa de imóveis, diálogo realizado com a Fabi, chatbot da Imobiliária Brognoli.

## 4.2 Coleta de dados

O foco da busca das imagens, para geração dos conjuntos de dados, é em imagens coloridas de fachadas de imóveis residenciais, mais precisamente casas. Por questões de exequibilidade o desenvolvimento ficou limitado ao uso de imagens da fachada, podendo ser estendido para outros tipos de imagens futuramente.

A busca por conjuntos de imagens começou em pesquisas nos mais populares bancos de dados disponíveis (Kaggle, ImageNet, LabelMe) e em pesquisas online. Tendo em vista a dificuldade em encontrar um conjunto adequado ao propósito deste trabalho, o qual seria um conjunto de imagens de fachadas de imóveis residenciais rotulado pela sua característica estrutural aparente, decidiu-se optar pela não utilização de um conjunto de dados pré-selecionado e rotulado e sim pela criação do mesmo. Desta forma, ao final do trabalho, foram gerados quatro conjuntos rotulados utilizados para treinamento do classificador.

A seleção das imagens, dentro dos conjuntos pré-selecionados, e posteriormente a rotulação de cada imagem foram realizadas de forma totalmente manual pelo autor.

O primeiro conjunto é constituído de imagens disponibilizadas no conjunto de dados de casas do repositório da ImageNet [16], o qual consta mais de 1200 imagens de diversas categorias referentes a casa. Após a separação das imagens do conjunto em imagens úteis, fachadas de imóveis residenciais, e não úteis, restaram 417 imagens para a utilização nos treinamentos.

O segundo conjunto é constituído de imagens retiradas do dataset Indoor Scene Recognition o qual foi criado para o artigo [17]. Neste conjunto consta 15620 imagens de 67 categorias de cenários diferentes. Após análise, foram selecionadas 789 imagens representando fachadas de imóveis residenciais.

O terceiro conjunto de dados foi criado a partir de imagens cedidas pela Imobiliária Brognoli, pertencente ao Grupo Brognoli, localizada em Florianópolis, Santa Catarina, do seu banco de dados. O uso das imagens foi autorizado pelo Grupo Brognoli. Em seu banco de dados constam mais de 40 mil imóveis cadastrados, porém, como este trabalho visa classificar apenas imagens de fachadas de imóveis residenciais, restaram cerca de 6 mil imóveis disponíveis. Devido ao alto número de imagens relacionadas a cada imóvel, mais de 60 mil imagens para 2,5 mil imóveis, e o tempo necessário para filtrar as imagens corretas, de fachadas, optou-se por não utilizar todos os cadastros disponíveis. Desta maneira, após selecionar e organizar as imagens, o conjunto alojou 1197 imagens classificadas.

A partir da fusão dos conjuntos dois e três formou-se o quarto conjunto, o qual consta com 1768 imagens, de maneira a obter um conjunto de dados com um volume maior de imagens e que represente de maneira mais igualitária a distribuição entre as classes.

## 4.2.1 Pré-processamento

A etapa de pré-processamento dos dados é de suma importância para o treinamento, podendo diversas técnicas serem utilizadas. Para implementação foi utilizado a função `ImageDataGenerator()` da biblioteca Keras [18]. A seguir são descritos e exemplificados as técnicas utilizadas nos processos de treinamento dos modelos de CNN.

### 4.2.1.1 Ajuste de escala

Cada pixel de uma imagem digital colorida, na representação matemática das cores no espaço de cor RGB, é constituído por 3 valores de 0 a 255. Para fins de treinamento dos classificadores realizou-se a normalização dos valores dos pixels para a escala de 0 até 1. Esse processo é realizado tanto para o conjunto de dados de treino como para o de validação e teste.

### 4.2.1.2 Ajuste de tamanho

Como o conjunto de dados pode ter imagens de diversas fontes, é necessário realizar o redimensionamento das imagens, o tamanho escolhido foi de 128 pixels de altura por 128 pixels de largura. A escolha do tamanho padrão da imagem foi obtida por meio de experimentos, os quais se deram pelo treinamento de uma rede CNN para diferentes tamanhos de imagens, a definição das dimensões utilizadas foi escolhida pela melhor taxa de acerto resultante do treinamento.

### 4.2.1.3 Data augmentation

O processo de data augmentation consiste em utilizar técnicas para digitalmente expandir o seu conjuntos de dados utilizando as mesmas imagens já disponíveis adicionando cópias ligeiramente modificadas, podendo assim, ajudar seu modelo a reduzir o overfitting durante o treinamento. As técnicas utilizadas são descritas e exemplificadas abaixo:

- Rotação: consiste em rotacionar a imagem horizontal, de 0 a 30 graus, em valores aleatórios dentro de uma faixa pré estabelecida, conforme demonstrado na Figura 12.

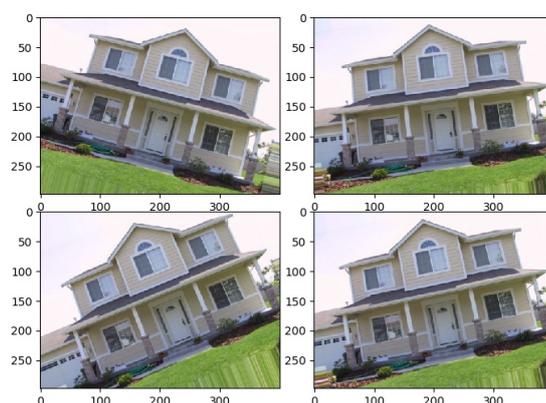


Figura 12 – Exemplo de imagens rotacionadas.

- Zoom: consiste em ampliar a imagem, conforme demonstrado na Figura 13.

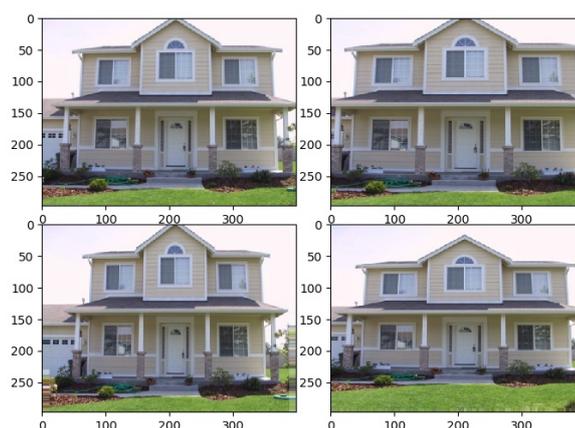


Figura 13 – Exemplo de imagens com zoom aplicado.

- Deslocamento em largura: consiste em transladar a imagem verticalmente em um valor aleatório, pré estabelecido, dentro de uma faixa de valores, fração de sua largura, conforme demonstrado na Figura 14.

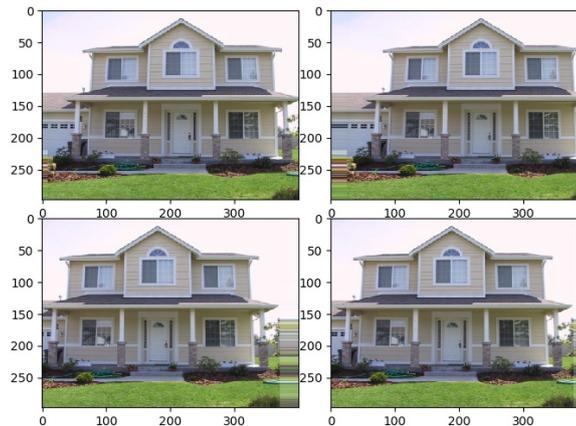


Figura 14 – Exemplo de imagens deslocadas em largura.

- Deslocamento em altura: consiste em transladar a imagem horizontalmente em um valor aleatório, pré estabelecido, dentro de uma faixa de valores, fração de sua altura, conforme demonstrado na Figura 15.

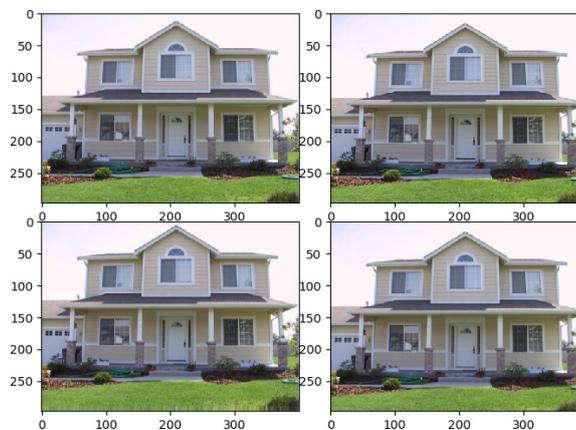


Figura 15 – Exemplo de imagens deslocadas em altura.

### 4.2.2 Manual Data Labelling

A rotulação do conjunto de dados foi realizada manualmente, ou seja, a partir da análise individual e reconhecimento das características de cada imagem foi atribuído o rótulo a imagem da classe selecionada.

Foram três as classes escolhidas para representar as características da fachada de um imóvel residencial, a saber: alvenaria, madeira e tijolo à vista. Essas classes foram obtidas por meio da observação das imagens nos conjuntos de dados criados.

A classe de imóvel residencial de alvenaria, Figura 16, não apresenta em quantidade significativa muitas linhas horizontais quanto verticais em sua fachada, sendo assim, uma estrutura mais uniforme, podendo apresentar variação de cores.



Figura 16 – Imagem modelo para classificação de imóveis como de alvenaria.

Já a classe de imóvel residencial de madeira, Figura 17, demonstra linhas horizontais mais demarcadas, de aparência listrada.



Figura 17 – Imagem modelo para classificação de imóveis como de madeira.

Por sua vez, a classe de imóvel residencial de tijolo a vista, Figura 18, apresenta linhas horizontais e verticais bem marcadas possuindo cores padrões (com algumas alterações de tonalidade).



Figura 18 – Imagem modelo para classificação de imóveis como de tijolo a vista.

Durante o processo de rotulação dos conjuntos de dados foram encontradas algumas dificuldades para a classificação das imagens de alguns imóveis, podendo ser: pela dificuldade de visualização da imagem (imagem com baixa qualidade ou muito pequena), por objetos obstruindo a fachada do imóvel (árvores, muros), pelo imóvel apresentar características pertencentes a mais de uma classe como demonstrado em Figura 19. Para contornar essas dificuldades houve o descarte de imagens impróprias e foi realizada análise visual mais minuciosa da imagem para que a partir das características mais presentes no imóvel rotular de maneira adequada.



Figura 19 – Exemplos de imagens de imóveis que possuem características de mais de uma classe.

### 4.3 Abordagens para solução do problema

O propósito central deste trabalho é criar um classificador capaz de ler imagens residenciais e com base em suas características predizer sua classe dentre um conjunto pré determinado de classes, desta forma, por meio da revisão bibliográfica efetuada e descrita no Capítulo 3, foram 3 as abordagens escolhidas como possíveis soluções deste problema.

Como ponto de partida foi escolhido utilizar técnicas de visão de computacional, mais precisamente extração de features, e posteriormente algoritmos de aprendizado de máquina para tarefa de classificação, de maneira a criar uma boa base de comparação para futuros resultados, tendo em vista a menor quantidade de poder computacional necessária em relação aos modelos de Deep Learning hoje em dia empregados. Em seguida, foi realizada a criação de CNNs específica para o problema e por último, foram testadas arquiteturas de CNN já conhecidas na literatura por meio da técnica de aprendizado de máquina chamada Transfer Learning. As abordagens citadas acima são descritas com mais detalhes nas próximas subseções.

### 4.3.1 Extração de Features mais utilização de algoritmos tradicionais

O primeiro passo para realizar esta abordagem é a computação dos descritores SIFT [13] e GIST [14] para o banco de imagens selecionado.

Para o descritor SIFT foi utilizado funções da biblioteca OpenCV [19] (Open Source Computer Vision), foi preciso determinar o número de features, características, extraídas de cada imagem, para utilizá-las nos testes de classificação, foi necessário realizar diversos experimentos para encontrar um número mínimo de características o qual todas as imagens do conjunto possuem. Após encontrado esse valor, foram extraídos os descritores de todas as imagens do conjunto e salvos em um arquivo .csv. Cada linha do arquivo representa uma imagem e nas colunas um valor referente ao descritor obtido em exceção da última coluna o qual foi adicionado o rótulo da imagem, necessário para o treinamento posterior dos dados.

Para o descritor GIST foi utilizado algoritmos em Python reimplementados do código em Matlab descrito em [20], desta maneira, não foi necessário encontrar o número de features, pois o tamanho do vetor descritor de todas as imagens se mantém o mesmo. O processo de arquivamento dos descritores ocorreu da mesma maneira que o do SIFT.

O treinamento dos dados obtidos pelos descritores foi realizado no software Weka [21] (Waikato Environment for Knowledge Analysis), o qual é uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados primeiramente desenvolvido pela Universidade de Waikato, Nova Zelândia, e mediante o uso dela foram testados os seguintes algoritmos de classificação presentes no software:

- RandomForest: classe para gerar uma floresta de árvores aleatórias, descrito em [22];
- J48: classe para gerar uma árvore de decisão C4.5, descrito em [23];
- MultiClassClassifier: classe que gera um metaclassificador para lidar com conjunto de dados multiclasse [21];

- `MultiClassClassifierUpdateable`: classe que gera um metaclassificador para lidar com conjunto de dados multiclasse, este classificador também é capaz de aplicar códigos de correção de erro na saída para melhorar a precisão [21];
- `IterativeClassifierOptimizer`: classe que otimiza o número de iterações de um classificador iterativo [21].

### 4.3.2 CNN específica para o problema

O processo de criação da CNN ocorreu a partir de um modelo base composto de 2 camadas convolucionais, 2 camadas de max pooling e 2 fully connected layers. A partir deste foram gerados, por meio de testes empíricos, diversos outros modelos podendo se diferenciar nos seguintes aspectos:

- Números de camadas;
- Números de features para cada camada;
- Tamanhos de filtros distintos;
- Funções de ativação;
- Otimizadores.

Dentre os modelos avaliados, foram 2 os modelos de arquiteturas, que apresentaram os melhores resultados em testes tendo desempenhos semelhantes, escolhidos para apresentar neste trabalho.

O primeiro modelo é composto por 4 camadas convolucionais, 2 camadas de max pooling e 2 fully connected layers, a Figura 20 representa o diagrama esquemático da CNN proposta e na tabela Tabela 1 constam os parâmetros de cada camada.

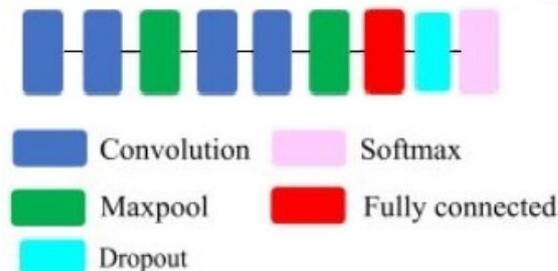


Figura 20 – Diagrama esquemático do primeiro modelo proposto.

O primeiro modelo gerado conta com cerca de 8,4 milhões de parâmetros treináveis.

Tabela 1 – Descrição dos parâmetros para o primeiro modelo de CNN proposto.

Layer	Feature Map	Size	Kernel Size	Stride	Activation
Image	1	128x128x3	-	-	-
Convolutional	16	128x128x16	3x3	1	ReLU
Convolutional	16	128x128x16	3x3	1	ReLU
Max-Pooling	16	64x64x16	2x2	2	ReLU
Convolutional	32	64x64x32	3x3	1	ReLU
Convolutional	32	64x64x32	3x3	1	ReLU
Max-Pooling	16	32x32x32	2x2	2	ReLU
Flatten	-	32768	-	-	-
Fully Connected	-	256	-	-	Sigmoid
Dropout	-	256	-	-	-
Fully Connected	-	3	-	-	Softmax

O segundo modelo é composto por 3 camadas convolucionais, 3 camadas de max pooling e 2 fully connected layers, a Figura 21 representa o diagrama esquemático da CNN proposta e na Tabela 2 constam os parâmetros de cada camada.

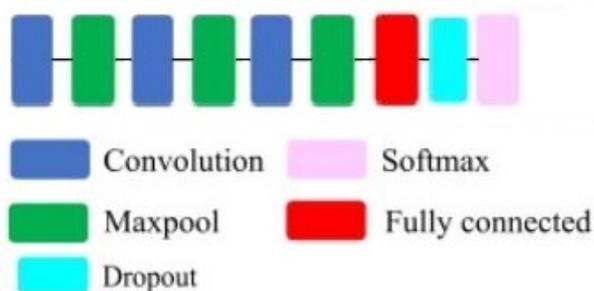


Figura 21 – Diagrama esquemático do segundo modelo proposto.

O segundo modelo gerado conta com cerca de 1,6 milhões de parâmetros treináveis.

Tabela 2 – Descrição dos parâmetros para o segundo modelo de CNN proposto.

Layer	Feature Map	Size	Kernel Size	Stride	Activation
Image	1	128x128x3	-	-	-
Convolutional	8	128x128x8	3x3	1	ReLU
Max-Pooling	8	64x64x8	2x2	2	ReLU
Convolutional	16	64x64x16	3x3	1	ReLU
Max-Pooling	16	32x32x16	2x2	2	ReLU
Convolutional	24	32x32x24	3x3	1	ReLU
Max-Pooling	24	16x16x24	2x2	2	ReLU
Flatten	-	6144	-	-	-
Fully Connected	-	256	-	-	Sigmoid
Dropout	-	256	-	-	-
Fully Connected	-	3	-	-	Softmax

Todos os algoritmos foram desenvolvidos utilizando a linguagem de programação Python e as bibliotecas Keras e TensorFlow, devido ao fato de serem de código aberto, bem documentadas, de fácil uso e com suporte ao uso de GPUs para o treinamento.

### 4.3.3 CNN usando transfer learning

Aprendizado por transferência é uma técnica de aprendizado de máquina em que um modelo já treinado para uma tarefa é reutilizado para iniciar outro modelo o qual irá realizar uma tarefa similar. As arquiteturas dos modelos de CNN selecionados foram as seguintes:

- VGG16: modelo constituído por 13 camadas convolucionais e 3 camadas totalmente conectadas, utilizando pilhas de pequenos filtros, de kernel 3x3 e stride 1, assim, aumentando a profundidade da rede e contribuindo ao aprendizado de características mais complexas, descrita em [24].

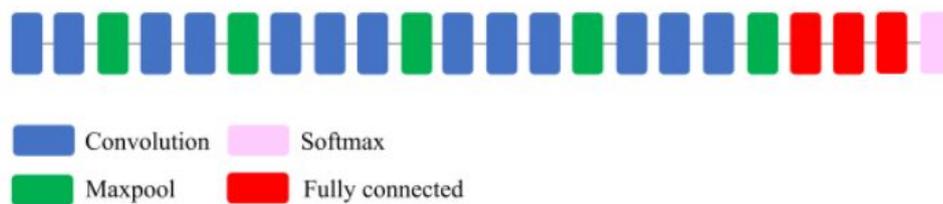


Figura 22 – Diagrama esquemático do modelo VGG16, imagem retirada de [24].

- VGG19: modelo semelhante a VGG16 porém com 3 camadas convolucionais a mais, assim, aumentando ainda mais a profundidade e complexidade da rede, descrita em [24].

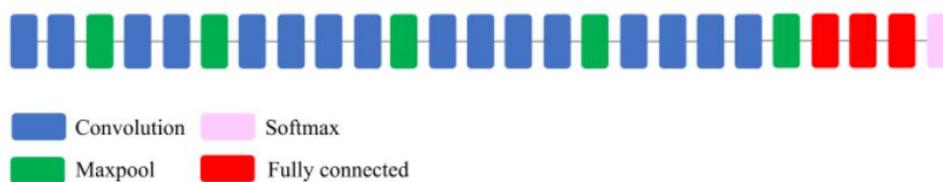


Figura 23 – Diagrama esquemático do modelo VGG19, imagem retirada de [24].

- ResNet50: modelo constituído de 50 camadas de profundidade, sendo 49 camadas convolucionais e 1 fully connected layer, na qual são empregados módulos de aprendizado residual entre as pilhas de filtros, descrita em [25].



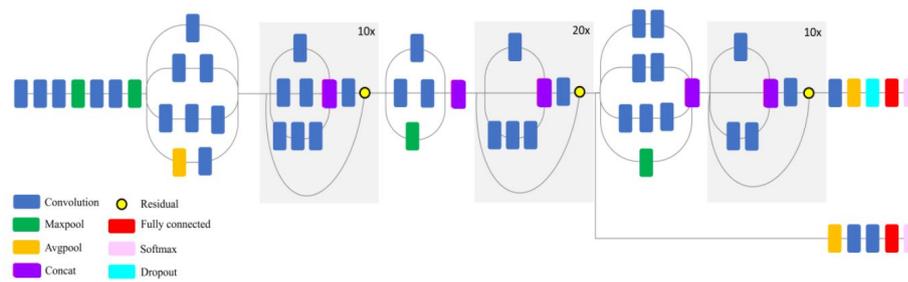


Figura 26 – Diagrama esquemático do modelo InceptionResNetV2, imagem retirada de [24].

Todas as arquiteturas foram implementadas em Python a partir de funções da biblioteca Keras, a qual carregava o modelo pré-treinado. Vale ressaltar que é necessário realizar ajustes nas arquiteturas para adaptá-las ao problema proposto, como por exemplo, os modelos vem pré-setados para o problema de classificação da ImageNet, que consiste na classificação de 1000 objetos, assim, tendo a camada Softmax tamanho 1000. Como a proposta é classificação de imóveis residenciais em 3 classes, essa camada foi atualizada para o valor adequado. Os demais parâmetros das redes foram mantidos.

## 4.4 Resumo dos experimentos e algoritmos testados

Neste capítulo, apresentou-se o processo da criação do conjunto de dados e os modelos de arquiteturas propostos para classificação plana multirrotulo do problema proposto. Os experimentos começaram pela extração de características por meio de descritores, SIFT e GIST, e seu treinamento utilizando os algoritmos de classificação do software Weka, em seguida foram desenvolvidos e avaliados modelos de CNN, elaborados pelo autor, para realização da tarefa classificação e por último foram adaptadas arquiteturas de CNN conhecidas por meio do uso da metodologia de Transfer Learning. No próximo capítulo são apresentados os resultados obtidos para cada experimento realizado utilizando todos os modelos descritos neste capítulo.

# 5 Experimentos e Resultados

Neste capítulo são apresentados e discutidos os resultados obtidos pelos modelos propostos e pelas demais arquiteturas citadas no capítulo anterior. A análise dos resultados é baseada nos resultados de precisão para o conjunto de treinamento, precisão para o conjunto de teste e a razão obtida entre ambos esses resultados.

## 5.1 Descrição dos Experimentos, Conjuntos de Dados, Parâmetros e Métricas

### 5.1.1 Cross-validation

Cross-validation (validação cruzada), é uma técnica utilizada para avaliar modelos de aprendizado de máquina em um conjunto de dados limitado com base no procedimento de reamostragem.

Este método consiste em separar o conjunto de dados em  $k$  grupos e, assim, realizar o treinamento do modelo de aprendizado de máquina  $k$  vezes para alterar o grupo escolhido como conjunto de teste nessas interações, de maneira a utilizar todo o conjunto de dados, em iterações diferentes, como treinamento e também como teste, assim, possibilitando resultados mais confiáveis e livres de viés gerado pela separação das partições de treinamento, validação e teste.

Segue o exemplo na Figura 27, em que o conjunto de dados é dividido em 10 grupos e é separado 1 grupo para teste, 1 grupo para validação e 8 grupos para treinamento em cada iteração.

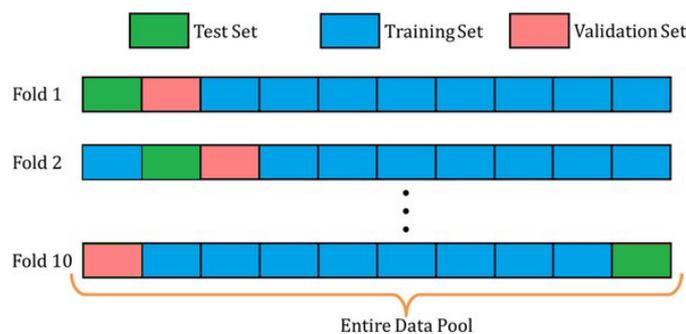


Figura 27 – Cross-validation, exemplo com número de pastas igual a 10.

Neste trabalho foi escolhido  $k=5$ , ou seja, o conjunto de dados dividido em 5 grupos, sendo 1 grupo para teste, 1 grupo para validação e 3 grupos para treinamento de maneira a ficar 20% para teste, 20% para validação e 60% para treinamento do conjunto de dados.

### 5.1.2 Parâmetros de aprendizado das redes

Na Tabela 3, são mostrados os parâmetros utilizados para o treinamento das redes CNN.

Tabela 3 – Parâmetro escolhidos para os Modelos 1 e 2.

Parâmetro	Valor
Epochs	50
Batch Size	64
Learning rate	0,001
Optimizer	RMSprop

Epochs: número de iterações realizadas no conjunto de dados.

Batch size: número de exemplos usados para cálculo do gradiente da função custo em relação aos parâmetros do modelo.

Learning rate: Taxa de aprendizado da rede neural.

Optimizer: Otimizador.

### 5.1.3 Métrica para avaliação dos modelos de classificação

A acurácia é uma métrica utilizada para avaliar modelos de classificação sendo definida como:

$$Acurácia = \frac{\text{Número de previsões corretas}}{\text{Número total de previsões}} \quad (5.1)$$

O coeficiente de overfitting (CO) é a razão entre a acurácia do conjunto de teste sobre a acurácia do conjunto de treinamento.

$$\text{Coeficiente de overfitting} = \frac{\text{Acurácia do conjunto de teste}}{\text{Acurácia do conjunto de treinamento}} \quad (5.2)$$

Macro F1 Score é utilizado para avaliar a qualidade dos problemas com várias classes, no qual 1 é o melhor valor e 0 o pior.

$$\text{MacroF1Score} = \frac{1}{N} * \sum_{i=0}^N F1_i \quad (5.3)$$

$$F1 = \frac{2 * (\text{Acurácia} * \text{Recall})}{(\text{Acurácia} + \text{Recall})} \quad (5.4)$$

$$\text{Recall} = \frac{\text{Número de verdadeiro positivos}}{(\text{Número de verdadeiro positivos} + \text{Número de falso negativos})} \quad (5.5)$$

Onde  $i$  é o índice da classe e  $N$  o número de classes.

### 5.1.4 Conjunto de dados

Os conjuntos de dados utilizados para realizar os testes com os classificadores foram os mencionados na Seção 4.2.

O primeiro conjunto de dados conta especificamente com 417 imagens de fachadas residências, distribuídas segundo a Tabela 4:

Tabela 4 – **Distribuição das imagens entre classes para o primeiro conjunto de dados.**

Classe	Distribuição(%)
Alvenaria	34
Madeira	33
Tijolo a vista	33

Para o segundo conjunto de dados, que contém 789 imagens, não apresenta a mesma equivalência na distribuição das imagens entre as classes, como demonstra a Tabela 5:

Tabela 5 – **Distribuição das imagens entre classes para o segundo conjunto de dados.**

Classe	Distribuição(%)
Alvenaria	28
Madeira	41
Tijolo a vista	31

No terceiro conjunto de dados constam 1197 imagens, com distribuição descrita na Tabela 6:

Tabela 6 – **Distribuição das imagens entre classes para o terceiro conjunto de dados.**

Classe	Distribuição(%)
Alvenaria	49
Madeira	20
Tijolo a vista	31

E por último, o quarto conjunto de dados contém 1768 imagens no total distribuídos conforme apresentado na Tabela 7:

Tabela 7 – **Distribuição das imagens entre classes para o quarto conjunto de dados.**

Classe	Distribuição(%)
Alvenaria	33
Madeira	32
Tijolo a vista	35

## 5.2 Resultados experimentais

A disposição dos resultados dos experimentos é realizada de maneira a tratar os conjuntos de dados, primeiramente, de maneira separada a fim de mostrar os resultados obtidos por cada arquitetura e a abordagem testada em cada conjunto de dados e, posteriormente, é realizado a análise dos dados obtidos.

O primeiro resultado obtido é referente à primeira abordagem para solução do problema, extração de Features mais a utilização de algoritmos tradicionais de classificação. Assim, são apresentados os dados de acurácia obtidos pelos classificadores citados na Seção 4.3.1 com os dados obtidos pelos descritores SIFT e GIST.

Os próximos resultados, referentes a segunda (Seção 4.3.2) e terceira (Seção 4.3.3) abordagens, representam a utilização de redes CNN para classificação de imagens. Desta maneira, são apresentados os dados de acurácia para os conjuntos de treinamento e teste para cada classificador e para treinamento utilizando e não utilizando data augmentation.

### 5.2.1 Resultados obtidos para o Primeiro Conjunto de Dados

A Tabela 8 representando os resultados obtidos pelos classificadores para os dados dos descritores SIFT e GIST:

Tabela 8 – **Resultados obtidos para a primeira abordagem, Extração de Features mais utilização algoritmos tradicionais, referentes ao primeiro conjunto de dados.**

Descritor / Classificador	Treinamento(%)	Teste(%)	CO	Enviesado
GIST / MultiClassClassifier	100	39,07	0,39	Não
GIST / MultiClassClassifierUpdateable	100	36,18	0,36	Não
GIST / RandomForest	100	46,29	0,45	Não
GIST / IterativeClassifierOptimizer	33,57	33,57*	-	Sim
SIFT / MultiClassClassifier	100	35,5	0,35	Não
SIFT / MultiClassClassifierUpdateable	100	33,09	0,33	Não
SIFT / RandomForest	100	35,71	0,35	
SIFT / IterativeClassifierOptimizer	33,57*	33,57*	-	Sim

Tabela 9 representando os dados obtidos pelos redes CNN utilizando Data augmentation:

Tabela 9 – **Resultados obtidos para as arquiteturas de CNN propostas referentes ao primeiro conjunto de dados utilizando Data Augmentation.**

Modelo	Treinamento(%)	Teste(%)	Macro F1 score	CO	Enviesado
Modelo 1	49,80	44,58	0,35	0,89	Não
Modelo 2	50,67	37,87	0,28	0,74	Não
InceptionResNetV2	65,38	48,64	0,27	0,74	Não
InceptionV3	35,48	33,80*	0,19	-	Sim
Vgg19	34,29	30,68*	0,15	-	Sim
Vgg16	34,19	31,44*	0,17	-	Sim

Tabela 10 representando os dados obtidos pelos redes CNN não utilizando Data augmentation:

Tabela 10 – **Resultados obtidos para as arquiteturas de CNN propostas referentes ao primeiro conjunto de dados sem utilizar Data Augmentation.**

Modelo	Treinamento(%)	Teste(%)	Macro F1 score	CO	Enviesado
Modelo 1	100	37,13	0,32	0,37	Não
Modelo 2	96,79	34,54	0,33	0,36	Não
InceptionResNetV2	65,15	40,07	0,26	0,62	Não
InceptionV3	42,03	32,39	0,19	0,77	Não
Vgg19	36,84	29,49	0,21	0,80	Não
Vgg16	36,20	29,24	0,19	0,81	Não

## 5.2.2 Resultados obtidos para o Segundo Conjunto de Dados

Tabela 11 representando os resultados obtidos pelos classificadores para os dados dos descritores SIFT e GIST:

Tabela 11 – **Resultados obtidos para a primeira abordagem, Extração de Features mais utilização algoritmos tradicionais, referentes ao segundo conjunto de dados.**

Descritor / Classificador	Treinamento(%)	Teste(%)	CO	Enviesado
GIST / MultiClassClassifier	100	39,41	0,39	Não
GIST / MultiClassClassifierUpdateable	99,77	45,11	0,45	Não
GIST / RandomForest	100	50,06	0,50	Não
GIST / IterativeClassifierOptimizer	61,11	49,94	0,82	Não
SIFT / MultiClassClassifier	100	35,86	0,36	Não
SIFT / MultiClassClassifierUpdateable	100	38,26	0,38	Não
SIFT / RandomForest	100	38,52	0,38	Não
SIFT / IterativeClassifierOptimizer	40,93*	40,93*	-	Sim

Tabela 12 representando os dados obtidos pelos redes CNN utilizando Data augmentation:

Tabela 12 – **Resultados obtidos para as arquiteturas de CNN propostas referentes ao segundo conjunto de dados utilizando Data Augmentation.**

Modelo	Treinamento(%)	Teste(%)	Macro F1 score	CO	Enviesado
Modelo 1	49,97	45,87	0,29	0,92	Não
Modelo 2	55,25	48,78	0,3	0,88	Não
InceptionResNetV2	62,73	46,13	0,27	0,74	Não
InceptionV3	46,00	41,19	0,22	0,90	Não
Vgg19	40,93	40,93*	0,19	-	Sim
Vgg16	40,93	40,93*	0,19	-	Sim

Tabela 13 representando os dados obtidos pelos redes CNN não utilizando Data augmentation:

Tabela 13 – **Resultados obtidos para as arquiteturas de CNN propostas referentes ao segundo conjunto de dados sem utilizar Data Augmentation.**

Modelo	Treinamento(%)	Teste(%)	Macro F1 score	CO	Enviesado
Modelo 1	87,69	42,95	0,31	0,49	Não
Modelo 2	79,71	40,17	0,34	0,50	Não
InceptionResNetV2	45,19	35,72	0,20	0,79	Não
InceptionV3	44,02	36,61	0,26	0,83	Não
Vgg19	38,61	39,55*	0,19	-	Sim
Vgg16	40,93	40,94*	0,19	-	Sim

### 5.2.3 Resultados obtidos para o Terceiro Conjunto de Dados

Tabela 14 representando os resultados obtidos pelos classificadores para os dados dos descritores SIFT e GIST:

Tabela 14 – **Resultados obtidos para a primeira abordagem, Extração de Features mais utilização algoritmos tradicionais, referentes ao terceiro conjunto de dados.**

Descritor / Classificador	Treinamento(%)	Teste(%)	CO	Enviesado
GIST / MultiClassClassifier	99,93	46,19	0,46	Não
GIST / MultiClassClassifierUpdateable	99,58	54,79	0,55	Não
GIST / RandomForest	99,93	55,22	0,55	Não
GIST / IterativeClassifierOptimizer	70,98	56,38	0,79	Não
SIFT / MultiClassClassifier	99,93	41,01	0,41	Não
SIFT / MultiClassClassifierUpdateable	99,62	44,27	0,44	Não
SIFT / RandomForest	99,93	48,28*	-	Sim
SIFT / IterativeClassifierOptimizer	49,89	48,45*	-	Sim

Tabela 15 representando os dados obtidos pelos redes CNN utilizando Data augmentation:

Tabela 15 – **Resultados obtidos para as arquiteturas de CNN propostas referentes ao terceiro conjunto de dados utilizando Data Augmentation.**

Modelo	Treinamento(%)	Teste(%)	Macro F1 score	CO	Enviesado
Modelo 1	61,09	56,21	0,32	0,92	Não
Modelo 2	60,95	56,47	0,31	0,93	Não
InceptionResNetV2	59,8	66,25	0,29	1,11	Não
InceptionV3	54,72	49,26	0,28	0,90	Não
Vgg19	48,86	48,87*	0,22	-	Sim
Vgg16	48,86	48,87*	0,22	-	Sim

Tabela 16 representando os dados obtidos pelos redes CNN não utilizando Data augmentation:

Tabela 16 – **Resultados obtidos para as arquiteturas de CNN propostas referentes ao terceiro conjunto de dados sem utilizar Data Augmentation.**

Modelo	Treinamento(%)	Teste(%)	Macro F1 score	CO	Enviesado
Modelo 1	90,94	48,45	0,34	0,53	Não
Modelo 2	76,93	49,70	0,32	0,65	Não
InceptionResNetV2	75,24	49,78	0,24	0,66	Não
InceptionV3	69,91	57,80	0,32	0,83	Não
Vgg19	48,86	48,87*	0,22	-	Sim
Vgg16	48,89	48,87*	0,22	-	Sim

#### 5.2.4 Resultados obtidos para o Quarto Conjunto de Dados

Tabela 17 representando os resultados obtidos pelos classificadores para os dados dos descritores SIFT e GIST:

Tabela 17 – **Resultados obtidos para a primeira abordagem, Extração de Features mais utilização algoritmos tradicionais, referentes ao quarto conjunto de dados.**

Descritor / Classificador	Treinamento(%)	Teste(%)	CO	Enviesado
GIST / MultiClassClassifier	99,95	39,13	0,39	Não
GIST / MultiClassClassifierUpdateable	98,87	48,01	0,49	Não
GIST / RandomForest	99,96	52,19	0,52	Não
GIST / IterativeClassifierOptimizer	62,60	49,48	0,79	Não
SIFT / MultiClassClassifier	99,95	35,23	0,35	Não
SIFT / MultiClassClassifierUpdateable	99,95	37,60	0,38	Não
SIFT / RandomForest	99,95	38,96	0,39	Não
SIFT / IterativeClassifierOptimizer	38,88	33,59*	-	Sim

Tabela 18 representando os dados obtidos pelos redes CNN utilizando Data augmentation:

Tabela 18 – **Resultados obtidos para as arquiteturas de CNN propostas referentes ao quarto conjunto de dados utilizando Data Augmentation.**

Modelo	Treinamento(%)	Teste(%)	Macro F1 score	CO	Enviesado
Modelo 1	53,46	50,89	0,31	0,90	Não
Modelo 2	55,89	51,63	0,32	0,91	Não
InceptionResNetV2	70,81	55,19	0,31	0,80	Não
InceptionV3	52,19	45,73	0,28	0,82	Não
Vgg19	34,87	33,48*	0,17	-	Sim
Vgg16	34,87	33,48*	0,17	-	Sim

Tabela 19 representando os dados obtidos pelos redes CNN não utilizando Data augmentation:

Tabela 19 – **Resultados obtidos para as arquiteturas de CNN propostas referentes ao quarto conjunto de dados sem utilizar Data Augmentation.**

Modelo	Treinamento(%)	Teste(%)	Macro F1 score	CO	Enviesado
Modelo 1	90,38	44,28	0,31	0,49	Não
Modelo 2	99,96	52,03	0,33	0,52	Não
InceptionResNet V2	81,89	57,62	0,30	0,70	Não
InceptionV3	80,93	56,67	0,33	0,70	Não
Vgg19	34,89	34,89*	0,17	-	Sim
Vgg16	34,89	34,95*	0,17	-	Sim

### 5.2.5 Análise de Dados

A primeira análise que pode ser realizada é referente aos descritores SIFT e GIST. O descritor GIST apresentou melhores resultados para todos os classificadores e conjuntos de dados de maneira geral, no entanto, todos os classificadores apresentaram overfitting a excessão do IterativeClassifierOptimizer, o qual teve resultados enviesados para os conjuntos de dados um e dois, porém o melhor desempenho para os conjuntos de dados três e quatro juntando as métricas de acurácia e coeficiente de overfitting.

Em sequência, podemos ressaltar a importância da utilização de Data Augmentation para o treinamento de redes CNN. De maneira geral reduziu o overfitting para o conjunto de treinamento, que para alguns casos pode chegar a 100%, e podendo aumentar a acurácia para o conjunto de teste de forma significativa, mais de 30% para o caso da rede InceptionResNet V2 para o segundo conjunto de dados.

Uma análise importante é sobre o quanto a quantidade de imagens no conjunto de dados influencia nos resultados e como a distribuição das imagens entre as classes interfere para resultados condizentes. Pelos dados obtidos é possível notar que há um aumento na acurácia do conjunto de teste a medida que os conjuntos de dados vão aumentando, como por exemplo o modelo de CNN Inception V3 que passou de 33% para o primeiro conjunto de dados para 51% de acurácia no quarto conjunto de dados, embora não seja

possível afirmar que a melhora deve-se apenas ao data augmentation. Essa comparação foi realizada pelo fato de ambos os conjuntos manterem a mesma distribuição de imagens entre as classes, caso que não acontece para o segundo e terceiro conjuntos, desta maneira, também podemos notar a importância deste dado, pois um modelo de machine learning pode estar obtendo um ótimo resultado de acurácia porém com péssimo desempenho na generalização das classes, o que acontece no caso da CNN InceptionResNet V2 para o terceiro conjunto, o qual apresenta desbalanceamento na distribuição das classes, tendo acurácia de 47% para a primeira classe, 18% para a segunda classe e 23% para terceira classe, assim, tornando a categorização do classificador tendenciosa para a classe com essa disparidade de imagens.

Entre os resultados obtidos pelas redes CNN, a que teve o melhor desempenho foi o modelo InceptionResNet V2, vale ressaltar o desempenho de ambas as redes elaboradas neste trabalho que, para o quarto conjunto de dados, tiveram acurácia semelhante, em torno de 51%, e cerca de 4% abaixo da InceptionResNet V2. Comportamento expressivo considerando a diferença de complexidade computacional e o número de parâmetros treináveis, 8,4 milhões e 1,6 milhões para o primeiro e segundo modelo respectivamente e 55,85 milhões para o modelo InceptionResNet V2. Ambos os modelos Vgg 16 e Vgg 19 não tiveram bons resultados para os conjuntos de dados apresentando resultados enviesados e muita dificuldade de generalização das imagens.

Analisando os resultados de ambas as abordagens apresentadas pelos critérios propostos conseguimos perceber que as redes neurais, de maneira geral, tem um desempenho melhor apresentando bons resultados de acurácia e não apresentando overfitting quando utilizado data augmentation, verificado pelo coeficiente de overfitting. Porém, as arquiteturas CNN são modelos mais complexos e que necessitam de um alto poder computacional, de maneira a fazer do descritor GIST, utilizando o classificador Iterative Classifier Optimizer, uma ótima opção para cenários menos robustos mostrando ser capaz de apresentar bons resultados.

Por fim, os resultados dos classificadores foram comparados através de um teste estatístico não paramétrico, para isso, optou-se pelo teste de Friedman que possibilita a comparação direta do desempenho de vários classificadores aplicados a vários conjuntos de dados. O teste de Friedman [26] é um teste estatístico não-paramétrico utilizado para determinar se existe ou não diferença estatística significativa entre os dados apurados. O teste realiza o cálculo de um valor p e compara-se a um nível de significância  $\alpha$ , geralmente no valor de 0,05, para determinar se há ou não diferenças estatísticas significativas entre os dados do teste. Para os resultados de testes obtidos, demonstrados na subseção acima, o valor de p obtido foi de 0,0009, ou seja, estatisticamente há diferença entre os resultados obtidos pelos diferentes modelos apresentados. Após realizado o teste de Friedman, é executado o teste de Nemenyi.

O teste de Nemenyi [26] é um teste post-hoc, isto significa, é um teste de comparação

múltipla utilizado após a aplicação de testes não paramétricos. O teste consiste em fazer comparações múltiplas par a par a fim de verificar qual dos fatores que diferem entre si, os resultados deste teste é apresentado pela Figura 28 por meio de um diagrama crítico.

O diagrama crítico é composto por um eixo horizontal enumerado no qual as classificações dos algoritmos testados são desenhadas. Os algoritmos são dispostos neste eixo considerando sua classificação, onde os melhores resultados são representados do lado esquerdo. A diferença crítica é mostrada acima do gráfico, de maneira a mostrar que todas as classificações médias dos algoritmos não diferem significativamente, para o nível de significância de  $p = 0,05$ , e são conectados por uma linha horizontal.

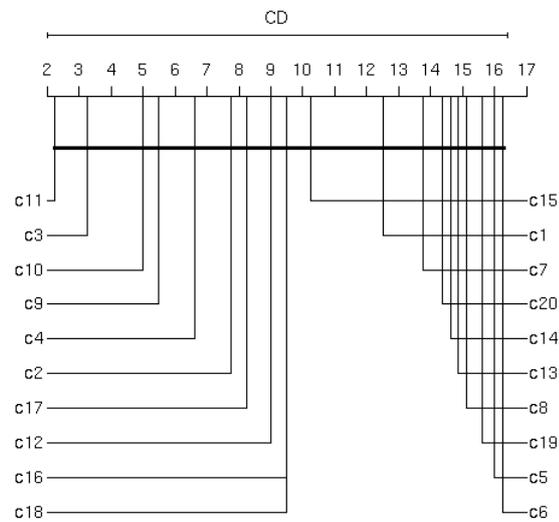


Figura 28 – Representação gráfica do resultado de teste post-hoc de Nemenyi.

- c1 - GIST com MultiClassClassifier
- c2 - GIST com MultiClassClassifierUpdateable
- c3 - GIST com RandomForest
- c4 - GIST com IterativeClassifierOptimizer
- c5 - SIFT com MultiClassClassifier
- c6 - SIFT com MultiClassClassifierUpdateable
- c7 - SIFT com RandomForest
- c8 - SIFT com IterativeClassifierOptimizer
- c9 - Modelo 1 - Utilizando Data augmentation
- c10 - Modelo 2 - Utilizando Data augmentation
- c11 - InceptionResNet V2 - Utilizando Data augmentation
- c12 - Inception V3 - Utilizando Data augmentation
- c13 - Vgg 19 - Utilizando Data augmentation
- c14 - Vgg 16 - Utilizando Data augmentation
- c15 - Modelo 1 - Não utilizando Data augmentation
- c16 - Modelo 2 - Não utilizando Data augmentation
- c17 - InceptionResNet V2 - Não utilizando Data augmentation
- c18 - Inception V3 - Não utilizando Data augmentation
- c19 - Vgg 19 - Não utilizando Data augmentation
- c20 - Vgg 16 - Não utilizando Data augmentation

Com base nos dados estatísticos apresentados, o teste de Friedman indicou que os classificadores são diferentes, apontando significância estatística, e no teste post-hoc de Nemenyi não foi possível apontar diferenciação entre os classificadores, pois todos ficaram dentro da distância crítica.

## 6 Conclusões

O presente trabalho apresentou o contexto da aplicação, o processo de coleta de dados, rotulação e pré-processamento de imagens para criação de um conjunto de dados apto para realizar tarefas de aprendizado de máquina e visão computacional. A partir deste contexto foi elaborada uma arquitetura para a aplicação do sistema oferecido, as abordagens propostas para a solução do problema e os resultados por elas obtidos.

A coleta de dados foi realizada de duas maneiras diferentes, sendo a primeira pela procura de conjunto adequados online e posteriormente pelo acesso ao banco de dados da Imobiliária Brognoli. Com isso, no processo de rotulação foi feita a escolha das classes as quais iriam catalogar as imagens, e para o pré-processamento foi mostrado as operações executadas nas imagens e o conceito de data augmentation.

Foram duas as abordagens propostas para o problema de classificação de fachadas de casas residências, a primeira consiste em utilizar descritores para capturar as características da imagem e a partir desses dados utilizar algoritmos tradicionais de classificação. A segunda abordagem se dá por meio da utilização de redes neurais convolucionais para classificação das imagens, para tal, foi proposto o uso tanto de redes CNN conhecidas quanto desenvolvidas especificamente para o problema abordado neste trabalho.

Os resultados alcançados foram obtidos por meio de testes experimentais em que todas as imagens foram expostas para todos os classificadores a fim de gerar uma solução mais robusta. Para tanto foram apuradas métricas, nas quais se baseiam as análises realizadas. Desta maneira, constata-se que os resultados obtidos pelas redes CNN, em sua maioria, apresentam o melhor desempenho para tarefa de classificação das imagens tendo sua maior acurácia com a rede InceptionResNet V2 usando transfer learning. Vale ressaltar o resultado obtido pelas redes CNN propostas por este trabalho, que tiveram rendimento um pouco abaixo dos máximos alcançados, porém sendo redes consideravelmente menos complexas e profundas. Destaca-se também o descritor GIST que junto ao classificador Iterative Classifier Optimizer obtiveram desempenho similar necessitando de um menor poder computacional para realização da tarefa.

Portanto, constata-se que os principais objetivos, descritos na Seção 1.3, deste trabalho foram alcançados e assim, se aprimorados, podem ser aplicados no setor imobiliário.

Como forma de aprimoramento do sistema proposto, pode-se citar para as seções relacionadas as imagens aumentar o conjunto de dados, adicionar conhecimentos arquitetônicos aos critérios de classificação e utilizar imagens de outras partes do imóvel para aplicar como características desejáveis no processo de filtragem dos imóveis resultantes.

Para as seções referentes a classificação e utilização de redes neurais pode-se apontar o aumento do número de classes para classificação das fachadas de casas residenciais,

tornar esse problema de classificação em Plana Multirrótulo (uma imagem poderá estar associada simultaneamente a dois ou mais rótulos), pelo fato de uma fachada residencial conter diversas características diferentes e testar o uso de redes neurais recorrentes como por exemplo a Long short-term memory (LSTM).

# Referências Bibliográficas

- 1 JARDIM, E.; LORRANY, L. Evolução recente do mercado de crédito no brasil. 2020. 14
- 2 ECONÔMICAS, F. I. de P.; BRASIL, R. de Imóveis do. Indicadores do registro imobiliário. 2021. 14
- 3 RAVIKUMAR, A. S. Real estate price prediction using machine learning. 2016. 15, 20
- 4 BAPPY, J. H. et al. Real estate image classification. Riverside, 2017. 15, 25
- 5 KOCH, D. et al. Visual estimation of building condition with patch-level convnets. 2018. 15, 26
- 6 ALPAYDIN, E. *Introduction to Machine Learning*. [S.l.]: MIT Press, 2009. 16
- 7 MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997. 16, 19
- 8 YALÇIN, O. G. *Applied Neural Networks with TensorFlow 2*. [S.l.]: Apress Media LLC, 2021. 16, 20, 22
- 9 FERRANDIN, M. Classificação hierárquica utilizando análise formal de conceitos. 2021. 18
- 10 HEATON, J. *AIFH, Volume 3: Deep Learning and Neural Networks*. [S.l.]: Heaton Research, Inc, 2015. 19
- 11 HAYKIN, S. *Neural Networks and Learning Machines Third Edition*. [S.l.]: Pearson Education, Inc., 2008. 19
- 12 GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. 21
- 13 SZELISKI, R. *Computer Vision: Algorithms and Applications 2nd Edition*. [S.l.]: Springer Science+Business Media, LLC, 2021. 24, 37
- 14 OUJAOURA, M. et al. Recognition of isolated printed tifinagh characters. 2014. 24, 37
- 15 SJÖQVIST, S.; RÅHLÉN, O. Image classification of real estate images with transfer learning. Stockholm, 2019. 27
- 16 IMAGENET. 2021. Disponível em: <<https://image-net.org/>>. 31
- 17 QUATTONI, A.; TORRALBA, A. Recognizing indoor scenes. 2009. 31
- 18 KERAS Documentation. 2021. Disponível em: <<https://keras.io/api/>>. 32
- 19 OPENCV library. 2021. Disponível em: <<https://opencv.org/>>. 37

- 
- 20 OLIVA, A.; TORRALBA, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. 2001. 37
- 21 FRANK MARK A. HALL, I. H. W. E. *The WEKA Workbench*. [S.l.: s.n.], 2016. 37, 38
- 22 BREIMAN, L. Random forests. 2001. 37
- 23 SALZBERG, S. L. C4.5: Programs for machine learning. J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. 37
- 24 MAHDIANPARI, M. et al. Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. 2018. 40, 41, 42
- 25 DEEP Residual Learning for Image Recognition. Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, 2015. 40
- 26 DEMsAR, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, JMLR.org, v. 7, 2006. 51