



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

Samuel Moreira Ransolin

**Detecção de Documentos Acadêmicos Falsificados: Uma Solução Baseada em
Aprendizado de Máquina**

Florianópolis
2026

Samuel Moreira Ransolin

**Detecção de Documentos Acadêmicos Falsificados: Uma Solução Baseada em
Aprendizado de Máquina**

Trabalho de Conclusão de Curso do Curso de
Graduação em Ciências da Computação do Centro
Tecnológico da Universidade Federal de Santa
Catarina para a obtenção do título de Bacharel em
Ciências da Computação.

Orientadora: Giovana Nunes Inocêncio, M.a.

Coorientador: Prof. Jean Everson Martina, Dr.

Florianópolis

2026

Samuel Moreira Ransolin

**Detecção de Documentos Acadêmicos Falsificados: Uma Solução Baseada em
Aprendizado de Máquina**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de
“Bacharel em Ciências da Computação” e aprovado em sua forma final pelo Curso de
Graduação em Ciências da Computação.

Florianópolis, 03 de julho de 2026.

Banca Examinadora:

Giovana Nunes Inocência, M.a.
Orientadora
Universidade Federal de Santa Catarina

Prof. Jean Everson Martina, Dr.
Coorientador
Universidade Federal de Santa Catarina

Lucas Machado da Palma, Dr.
Universidade Federal de Santa Catarina

Gabriel Estevam de Oliveira, M.e.
Universidade Federal de Santa Catarina

RESUMO

Nos últimos anos, no Brasil, o expressivo aumento do número de ingressantes, formandos e de instituições de ensino superior, trouxe consigo desafios relacionados à validação da autenticidade de certificados acadêmicos. Atualmente, o processo consiste na verificação de forma predominantemente manual, sujeita a erros e falhas, como a aceitação de documentos falsificados. Nesse contexto, surge a Jornada do Estudante, um sistema disponibilizado pelo Ministério da Educação (MEC), que oferece, entre outras funcionalidades, o acompanhamento de registros acadêmicos de estudantes por meio de uma rede distribuída permissionada, de forma a garantir que somente instituições de ensino digitalmente comprovadas possam registrar créditos e certificações. O presente trabalho de conclusão de curso revisita o estado da arte em detecção de documentos falsificados por aprendizado de máquina e desenvolve um protótipo de solução híbrida que, a partir da imagem digitalizada de documentos acadêmicos, encadeia o agrupamento em famílias de mesmo *template*, a geração de falsificações para treinamento e um detector forense supervisionado, capaz de localizar adulterações no nível do *pixel*, classificando cada exemplar conforme seu grau de legitimidade. Avaliado sobre a classe de falsificações mais realista, que reproduz adulterações sutis e localizadas em campos do documento, o detector conseguiu detectar 75% dos documentos adulterados, com precisão de 97% entre os exemplares apontados como fraudulentos. Ao integrar esse sistema à Jornada do Estudante, é possível validar automaticamente os documentos antes de seu registro em *blockchain*, aumentando significativamente a segurança e a confiabilidade do processo de credenciamento.

Palavras-chave: segurança da informação, detecção de fraude, *machine learning*, *clustering*, detecção de anomalias, análise forense de imagens

ABSTRACT

In recent years, the significant growth in the number of incoming students, graduates, and higher education institutions in Brazil has brought challenges related to validating the authenticity of academic certificates. Currently, this process relies on predominantly manual verification, which is prone to errors and failures, such as the acceptance of forged documents. In this context emerges the Jornada do Estudante, a system provided by the Ministério da Educação (MEC) that offers, among other features, the tracking of students' academic records through a permissioned distributed network, ensuring that only digitally verified educational institutions can register credits and certifications. This undergraduate thesis revisits the state of the art in the machine-learning-based detection of forged documents and develops a prototype of a hybrid solution that, from the digitized image of academic documents, chains together the clustering of documents into families sharing the same template, the generation of forgeries for training, and a supervised forensic detector capable of localizing tampering at the pixel level, classifying each instance according to its degree of legitimacy. Evaluated on the most realistic class of forgeries, which reproduces subtle, localized tampering in document fields, the detector could identify 75% of the tampered documents, with a precision of 97% among the instances it flagged as fraudulent. By integrating this system into the Jornada do Estudante, documents can be automatically validated before being recorded on the blockchain, significantly increasing the security and reliability of the accreditation process.

Keywords: information security, fraud detection, machine learning, clustering, anomaly detection, image forensics

LISTA DE FIGURAS

Figura 1 – Exemplo de Ruído Deixado por Adulteração de Imagem	15
Figura 2 – Representação de Aprendizado Supervisionado	16
Figura 3 – Representação de Aprendizado Não Supervisionado	17
Figura 4 – Representação de Aprendizado por Reforço	17
Figura 5 – Representação de Aprendizado Auto-Supervisionado	18
Figura 6 – Representação de Aprendizado Contrastivo	18
Figura 7 – Representação de um Neurônio Artificial	19
Figura 8 – Representação de uma Rede Profunda	20
Figura 9 – Visualização do Gradiente Descendente	21
Figura 10 – Visualização de uma Operação de Convolução	22
Figura 11 – Representação da Classificação de uma Imagem por uma Rede Convo- lucional	23
Figura 12 – Representação do Mecanismo de Atenção	26
Figura 13 – Representação de um Vision Transformer	27
Figura 14 – Representação de um Algoritmo de <i>Clustering</i> por Distância	30
Figura 15 – Representação do Algoritmo DBSCAN	30
Figura 16 – Representação do Agrupamento Aglomerativo	31
Figura 17 – Representação da Arquitetura de Validação de Diplomas por Kim	33
Figura 18 – Representação da Arquitetura de Fusão Multimodal por Jain; Wigington	37
Figura 19 – Resultado da Verificação de um Documento Autêntico	38
Figura 20 – Diagrama de Treinamento DINO	40
Figura 21 – Representação de MAC	42
Figura 22 – Representação de R-MAC	43
Figura 23 – <i>Pipeline</i> ASC-Former	44
Figura 24 – DBSCAN Sobre Extrações EfficientNet-B0	48
Figura 25 – DBSCAN Sobre Extrações DiT	49
Figura 26 – DBSCAN Sobre Extrações DINOv2	50
Figura 27 – Representação do Fluxo de Inferência	51
Figura 28 – Representação do Fluxo de Treino	52
Figura 29 – Detalhamento do Módulo de Construção de Dataset	53
Figura 30 – Amostras do <i>Dataset</i>	54
Figura 31 – Detalhamento do Módulo de Extração de Layout	55
Figura 32 – Detalhamento do Módulo de Agrupamento de Layout	56
Figura 33 – Amostras de Documentos do Mesmo Cluster de Primeira Ordem	57
Figura 34 – Amostras de Documentos do Mesmo Cluster de Segunda Ordem	57
Figura 35 – Amostras de Documentos Sinteticamente Falsificados	59
Figura 36 – Ilustração do Treinamento do Detector Forense	61

Figura 37 – Exemplo de Resultado de Inferência (no painel: o documento falsificado anonimizado; o mapa de adulteração; overlay do mapa de adulteração no documento; o rótulo de adulteração)	63
Figura 38 – Representação dos Clusters de Primeira Ordem	65
Figura 39 – Representação dos Clusters de Segunda Ordem	67
Figura 40 – Exemplos de Falsificações de Difícil e Fácil Detecção	69

LISTA DE TABELAS

Tabela 1 – Seleção do <i>linkage</i> e do limiar de corte do primeiro estágio	66
Tabela 2 – Estratificação em treino, validação e teste	70
Tabela 3 – Desempenho do detector a nível de documento, por classe	70
Tabela 4 – Localização do detector a nível de <i>pixel</i> , por classe	71
Tabela 5 – Resumo da inspeção visual do detector, com 25 amostras por tipo, indicando os documentos sinalizados e o escore médio	71

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVOS	11
1.1.1	Objetivo Geral	11
1.1.2	Objetivos Específicos	11
1.2	METODOLOGIA	12
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	FUNDAMENTOS DA FORENSE DIGITAL SOBRE IMAGENS	14
2.2	APRENDIZADO DE MÁQUINA	15
2.3	REDES NEURAIS PROFUNDAS	19
2.3.1	Redes Neurais Convolucionais	21
2.3.1.1	Reconhecimento Óptico de Caracteres (OCR)	23
2.3.2	Transformadores	24
2.3.2.1	Vision Transformers	27
2.3.3	Representação por Vetores de Características	28
2.4	ALGORITMOS DE AGRUPAMENTO	29
2.4.1	Deteccção de Anomalias	31
3	TRABALHOS CORRELATOS	32
3.1	VISÃO GERAL	32
3.2	INTEGRANDO APRENDIZADO DE MÁQUINA À <i>BLOCKCHAIN</i> PARA PREVENÇÃO E DETECÇÃO DE FRAUDES EM DIPLOMAS	32
3.3	APLICANDO ANÁLISE MULTIMODAL A DOCUMENTOS	35
3.4	TRATANDO A AUTENTICAÇÃO DE DOCUMENTOS ACADÊMICOS COMO UM PROBLEMA DE AGRUPAMENTO	37
4	TRABALHOS BASE	39
4.1	APRENDIZADO DE CARACTERÍSTICAS VISUAIS SEM SUPERVISÃO	39
4.2	AGREGAÇÃO REGIONAL DE CARACTERÍSTICAS VISUAIS	40
4.3	DETECÇÃO DE ADULTERAÇÃO DE TEXTO EM DOCUMENTOS	43
5	ARQUITETURA DA SOLUÇÃO	47
5.1	DA ABORDAGEM INICIAL À SOLUÇÃO ATUAL	47
5.2	VISÃO GERAL DA ARQUITETURA	51
5.3	TREINAMENTO	52
5.3.1	Construção do Dataset	52
5.3.2	Extração de Layout	54
5.3.3	Agrupamento de Layout	56
5.3.4	Geração de Falsificações	58
5.3.5	Treinamento do Detector Forense	60
5.4	INFERÊNCIA	62

6	RESULTADOS	64
6.1	CONFIGURAÇÃO EXPERIMENTAL E DATASET	64
6.2	DESCOBERTA DE TEMPLATES	64
6.3	GERAÇÃO DE FALSIFICAÇÕES	68
6.4	DETECTOR FORENSE SUPERVISIONADO	70
6.5	LIMITAÇÕES OBSERVADAS	71
7	CONCLUSÃO	73
	REFERÊNCIAS	74
	APÊNDICE A – ACESSO AO CÓDIGO-FONTE	79
	APÊNDICE B – ARTIGO	80

1 INTRODUÇÃO

No Brasil, entre 2013 e 2023, o número de matrículas de alunos na educação superior aumentou 36,2%, com uma média de crescimento anual de 3,2%. O número de concluintes acompanhou essa mesma tendência de crescimento, sendo que o ano de 2013 registrou cerca de 992 mil graduandos, enquanto 2023 terminou com mais de 1,3 milhões. Para acomodar essa demanda, existem 2580 entidades de ensino superior no país (BRASIL, 2024), das quais 87,8% são privadas. Essas estatísticas revelam um saldo extremamente positivo, mas também trazem à tona desafios que precisam ser superados, como a melhoria nos processos de regulação, supervisão e avaliação destas entidades por parte do Ministério da Educação do Brasil (MEC), temática que pretende ser explorada nesse trabalho.

Atualmente, a gerência, armazenamento e cuidado de documentos acadêmicos, como diplomas e históricos escolares, é responsabilidade da instituição de ensino que os emitem (MEC, 1978). Além disso, o próprio processo para a emissão desses documentos é burocrático, não computadorizado e sujeito a erros ou até mesmo fraudes, já que suas validações não possuem transparência ou redundância (Palma et al., 2019). Assim, a falta de modernização desses procedimentos deixam brechas que são conhecidas e utilizadas por agentes mal intencionados, possibilitando a criação de falsas instituições de ensino, especializadas na venda de pacotes que incluem históricos escolares falsificados, atas de colação de grau inexistentes e outros certificados contrafeitos, amparados em documentos oficiais adulterados, de forma a conferir aparência de legalidade a diplomas que, efetivamente, não têm qualquer base acadêmica real (Dias; Leal, 2022).

O comércio clandestino de diplomas falsos oferece certificados em diversas áreas e níveis, desde medicina até direito; desde a graduação até o doutorado, por valores que podem chegar a R\$100.000, tornando essa prática altamente lucrativa e atraente para fraudadores (Palma et al., 2019). Investigações recentes demonstram que quadrilhas estruturadas conseguem emitir dezenas de milhares de documentos forjados, comercializados em sites especializados, com suposta publicação em diários oficiais (Globo, 2025). Para além da corrupção, esse tipo de fraude compromete a confiança pública nas instituições de ensino e no mercado de trabalho: indivíduos sem qualificação adequada podem assumir funções críticas, enquanto diplomas legítimos perdem valor diante da insegurança sobre sua autenticidade (Mohammed; Nwobodo; Ekene, 2024).

Neste cenário, o MEC, em parceria com o Ministério da Economia, disponibiliza e desenvolve o sistema da Jornada do Estudante junto a Universidade Federal de Santa Catarina (UFSC, por meio do Laboratório Bridge e do Laboratório de Segurança em Computação), a Universidade Tecnológica Federal do Paraná e a Universidade Federal de Mato Grosso do Sul (MEC, 2022). Este sistema permite que alunos acompanhem seus dados estudantis durante toda a trajetória educacional, além da disponibilização de documentos acadêmicos pertinentes. Em conjunto a isso, existe a iniciativa para que se

torne uma plataforma conjunta para a emissão e registro destes certificados, unificando diplomas, históricos escolares, currículos e até mesmo dados regulatórios das instituições de ensino superior (RNP, 2023).

Para armazenar estes dados, o sistema da Jornada do Estudante utiliza uma *block-chain Hyperledger Fabric*, que aproveita características como descentralização da posse e imutabilidade dos registros, além da rastreabilidade às emissões. O projeto realiza o processamento dos dados em rede através de *smart contracts* e baseia-se em gestão de identidade forte, com certificados digitais que servem como base da identidade, de forma que somente entidades reconhecidas pelo projeto possam efetuar transações (Palma et al., 2019; RNP, 2023).

Ainda assim, hoje, a Jornada do Estudante não elimina o risco do registro de documentos falsificados, mas seu arcabouço permite o desenvolvimento de uma solução para este desafio. O presente Trabalho de Conclusão de Curso (TCC) trata da implementação e validação de um protótipo de *software* de inteligência artificial, que combina diferentes técnicas de aprendizado de máquina, capaz de identificar imagens editadas de certificados antes de sua inserção neste ambiente. Trata-se de uma investigação acadêmica independente, conduzida a partir de uma base de documentos autênticos cedida pela UFSC, dessa forma, ao integrar essa tecnologia à Jornada do Estudante, espera-se aprimorar o processo de registro e emissão de documentos acadêmicos no Brasil, de forma que o emprego dessa análise automatizada, no momento do registro de cada documento, possa garantir maior segurança e confiabilidade a estes procedimentos.

1.1 OBJETIVOS

Esta seção apresenta o objetivo geral e objetivos específicos deste trabalho.

1.1.1 Objetivo Geral

Desenvolver um protótipo de software capaz de classificar imagens de documentos acadêmicos quanto à suspeita de fraude, com base na integração de diferentes técnicas de aprendizado de máquina.

1.1.2 Objetivos Específicos

Para atingir o objetivo geral, os seguintes objetivos específicos devem ser cumpridos:

- Adquirir uma base de documentos acadêmicos digitalizados e autênticos;
- Desenvolver sistema para pré-processamento dos documentos, aplicando normalização e reconhecimento óptico de caracteres;
- Desenvolver sistema capaz de extrair impressões digitais de *layout* e agrupar os documentos em famílias de *templates*;

- Desenvolver sistema para a produção de diversos tipos de falsificações sintéticas com base nas famílias de *templates* originais, para o treinamento, teste e validação do protótipo;
- Desenvolver e treinar sistema capaz de detectar e apontar as falsificações fabricadas, que produza mapas de calor de adulteração por *pixel* e escores de suspeita por documento;
- Avaliar a capacidade de generalização do sistema, isto é, medir seu desempenho na detecção de falsificações sobre documentos e técnicas de adulteração não vistos durante o treinamento.

1.2 METODOLOGIA

A metodologia adotada neste trabalho tem carácter exploratório e experimental. Trata-se de uma pesquisa aplicada, voltada à construção e avaliação de um artefato de *software* para a detecção de fraudes em imagens de documentos acadêmicos. Grande parte do esforço concentrou-se na experimentação, isto é, diversas abordagens foram levantadas, testadas e descartadas até a concepção da solução final, cujo percurso é detalhado no Capítulo 5. A avaliação, por sua vez, apoia-se sobretudo em métricas de qualidade somadas a verificações visuais pontuais em etapas intermediárias.

As etapas realizadas ao longo do trabalho descrevem-se a seguir. Primeiramente, revisou-se a literatura, o que compreendeu a fundamentação teórica da conceituação, algoritmos e técnicas de aprendizado de máquina aproveitados; o levantamento de trabalhos correlatos; e o estudo dos trabalhos que servem de base direta à solução.

Em seguida, construiu-se a base de dados sobre a qual o trabalho operou, a partir de um acervo de documentos cedido pela UFSC, que passou por processo de filtragem a fim de remover cópias repetidas e preservar somente documentos de interesse. A partir desse conjunto, extraiu-se de cada documento um descritor de sua estrutura visual em representação computável, que foram utilizados para agrupar os documentos por similaridade, de modo a descobrir famílias de *templates* presentes no acervo original. A configuração desse agrupamento não foi arbitrada, mas decorreu de uma busca experimental, em que diferentes parâmetros foram avaliados por métricas de coesão dos grupos formados.

Com as famílias de documentos identificadas, os exemplos de falsificações necessários ao aprendizado do sistema foram gerados de forma sintética. Cada família de *template* forneceu documentos compatíveis entre si, sobre os quais se produziram falsificações de diferentes tipos, cada uma acompanhada da marcação exata das regiões editadas. O conjunto resultante foi dividido em treino, validação e teste, sob restrições que impedem que um mesmo documento ou família apareça em mais de um desses conjuntos. Dessa forma, sobre esse material, treinou-se um modelo de visão computacional para detectar as adulterações.

Por fim, avaliou-se esse detector por métricas de qualidade, como precisão, *recall* e F1, medindo seu desempenho separadamente para cada classe de falsificação. Incluiu-se um cenário voltado à capacidade de generalização, em que o sistema é confrontado com técnicas de adulteração ausentes do treino, cujos resultados são apresentados e discutidos no Capítulo 6.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda os conceitos teóricos necessários para a compreensão do presente trabalho.

2.1 FUNDAMENTOS DA FORENSE DIGITAL SOBRE IMAGENS

A forense digital de imagens é o campo que tem por objetivo verificar a integridade de uma imagem e detectar vestígios de manipulação. De acordo com Birajdar; Mankar (2013), as principais técnicas dividem-se em duas categorias, a abordagem ativa e a passiva. A ativa depende de um código de autenticação, como uma marca-d'água ou assinatura digital, inserido na imagem no momento de sua criação, o que pressupõe controle sobre o processo de geração do documento (Birajdar; Mankar, 2013). A abordagem passiva, ou cega, opera apenas sobre a imagem recebida, sem qualquer estampa do emissor nem conhecimento prévio sobre seu conteúdo (Birajdar; Mankar, 2013), e por isso é de interesse deste trabalho, em que a avaliação se dá sobre documentos acadêmicos que chegam ao sistema como imagens digitalizadas por *scanner* ou fotografadas, desprovidas de qualquer mecanismo de autenticação prévio. A premissa que sustenta a análise passiva é a de que uma falsificação, ainda que não deixe indícios visuais perceptíveis, perturba a regularidade estatística que a imagem deveria manter, introduzindo artefatos que denunciam a manipulação (Birajdar; Mankar, 2013).

As principais manipulações que recaem sobre uma imagem podem ser organizadas em alguns tipos principais. Técnica comum é a clonagem, ou *copy-move*, em que uma região da própria imagem é copiada e colada em outra posição, com o intuito de ocultar ou de duplicar algum elemento, assim, sua detecção exige a busca por padrões duplicados (Birajdar; Mankar, 2013). Distinta dela é a composição, ou *splicing*, em que fragmentos de uma ou mais imagens diferentes são inseridos na imagem alvo, trazendo consigo características que conflitam com as do documento que as recebe (Birajdar; Mankar, 2013). Também se destacam as operações de retoque, como ajustes de contraste, borramento e reamostragem por escala ou rotação, frequentemente aplicadas sobre as regiões manipuladas para suavizar as bordas e disfarçar os vestígios das demais alterações (Birajdar; Mankar, 2013). Transpostas ao contexto de um documento, essas operações correspondem a duplicar um campo, importar o valor de outro documento ou apagar e repintar um trecho do conteúdo, por exemplo.

Para além da natureza da manipulação, a forense passiva explora os rastros que o próprio processo de aquisição imprime na imagem (Birajdar; Mankar, 2013). Ao converter a luz em um arquivo, um dispositivo de captura, câmara ou *scanner*, executa uma cadeia de operações (como a interpolação de cores, a correção de gama e a compressão) que, somadas às imperfeições físicas do sensor, deixam em cada imagem um conjunto de artefatos fracos e muitas vezes humanamente imperceptíveis, característicos do aparelho e

do modelo que a geraram (Cozzolino; Verdoliva, 2018). Esses artefatos concentram-se no chamado ruído residual, o sinal semelhante a ruído que sobra quando se remove o conteúdo visível de uma cena, e funcionam como uma impressão digital da história de captura e compressão da imagem (Cozzolino; Verdoliva, 2018). O ponto central para a detecção é que tais artefatos são espacialmente regulares e aproximadamente uniformes ao longo de uma imagem autêntica, de modo que uma edição local rompe essa regularidade e produz, na assinatura de ruído, uma inconsistência que delata e localiza a região manipulada, mesmo quando nada é visível a olho nu (Cozzolino; Verdoliva, 2018), como exemplificado na Figura 1, à esquerda a imagem forjada, e à direita a extração desses artefatos, que evidencia a forja. Por, no fundo, se tratarem de anomalias estatísticas, o aprendizado de máquina tem se mostrado cada vez mais eficaz na análise forense (Qu et al., 2023).

Figura 1 – Exemplo de Ruído Deixado por Adulteração de Imagem



Fonte: (Cozzolino; Verdoliva, 2018)

A partir desses fundamentos, delimita-se o modelo de ameaça assumido neste trabalho. Parte-se do pressuposto de que o falsificador não produz um documento do zero, mas toma como base um documento legítimo e altera localmente apenas os campos de seu interesse, como nomes, notas e datas. Dessa forma, a maior parte do documento, incluindo selos da instituição emissora, o *layout* e o conteúdo textual geral, permanece autêntica, de modo que apenas regiões efetivamente editadas carregam pistas de adulteração. É sobre a detecção e a localização dessas regiões que se concentra a solução proposta.

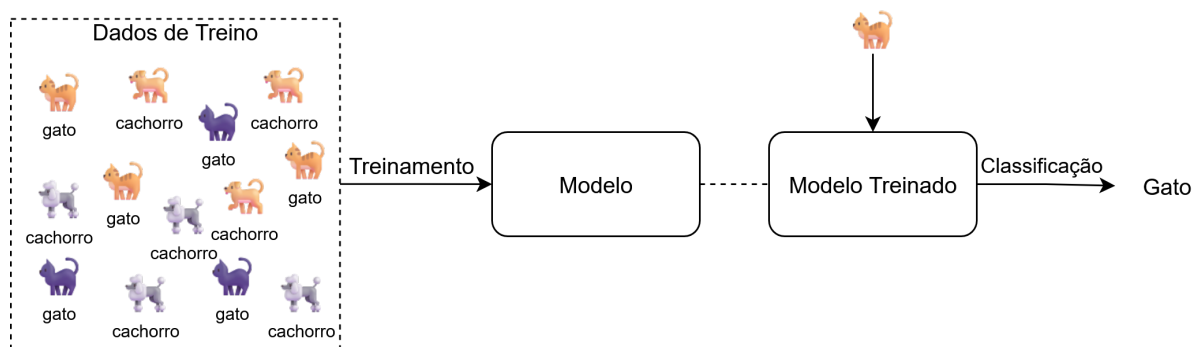
2.2 APRENDIZADO DE MÁQUINA

O aprendizado de máquina é um subcampo da inteligência artificial que tem por objetivo o desenvolvimento de algoritmos capazes de aprender e tomar decisões a partir de um conjunto de dados, sem que seja necessária a programação explícita para essas tarefas específicas (Dietterich, 2003). Fundamentalmente, esses sistemas buscam aprender padrões

em coleções de dados para, a partir da generalização desse conhecimento, realizar inferências sobre novas informações. Esse processo de aprendizado utiliza modelos matemáticos, principalmente estatísticos, que capturam relações complexas entre variáveis de entrada e saída através do ajuste de parâmetros internos, permitindo que a aplicação melhore seu desempenho conforme é exposta a mais dados (Sarker, 2021; Dietterich, 2003). Em geral, as técnicas de *machine learning* são categorizadas em três paradigmas principais: aprendizagem supervisionada, não supervisionada e por reforço (Sarker, 2021).

O aprendizado supervisionado caracteriza-se pela utilização de conjuntos de dados rotulados, onde tanto as entradas quanto as saídas desejadas são conhecidas durante o treinamento. Nesse paradigma, o algoritmo aprende através de exemplos, de forma a possibilitar tarefas como classificação – a atribuição de classes discretas aos dados, como ilustrado pela Figura 2 – e regressão – a predição de valores contínuos (Dietterich, 2003). Algoritmos clássicos dessa categoria incluem máquinas de vetores de suporte, redes neurais artificiais e métodos *ensemble* (Sarker, 2021).

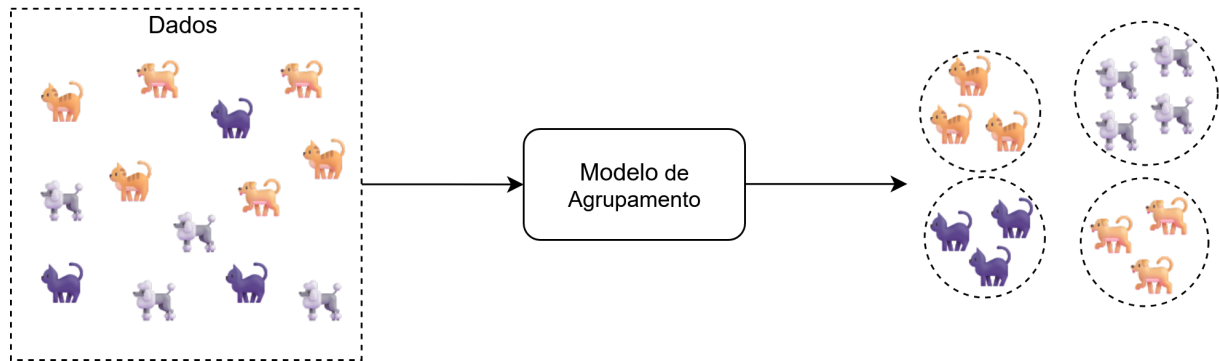
Figura 2 – Representação de Aprendizado Supervisionado



Fonte: o autor

O aprendizado não-supervisionado, por sua vez, opera sobre dados não rotulados, sem o conhecimento das saídas desejadas, e busca compreender a organização natural de um dado conjunto a partir da identificação de padrões intrínsecos. Essa abordagem engloba técnicas como agrupamento (*clustering*), ilustrado pela Figura 3, e detecção de anomalias (Sarker, 2021). Ainda, diferentes estratégias de aprendizado podem ser incorporadas, como algoritmos semi-supervisionados, utilizados quando um *dataset* tem poucos dados classificados, de forma a aproveitar a estrutura implícita do conjunto não categorizado para melhorar o desempenho do modelo (Sarker, 2021). O aprendizado por reforço, terceiro paradigma, em que um modelo aprende por tentativa e erro a partir de recompensas e penalidades do ambiente, não é empregado neste trabalho.

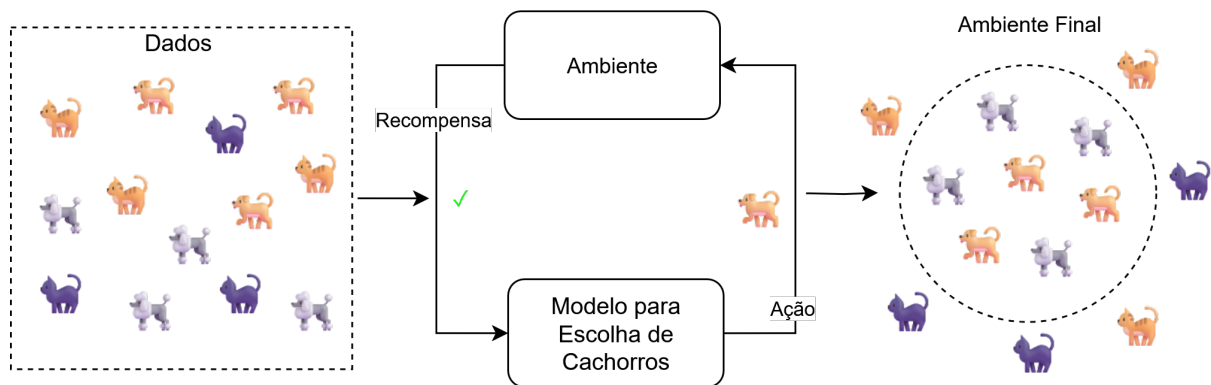
Figura 3 – Representação de Aprendizado Não Supervisionado



Fonte: o autor

Adicionalmente, o aprendizado por reforço representa um paradigma distinto onde um modelo aprende através de interações com um ambiente, sendo recompensado ou penalizado com base em suas ações, de forma que gradualmente desenvolva estratégias ótimas (Sarker, 2021). A Figura 4 ilustra o exemplo de um modelo com recompensas positivas para a identificação de cães.

Figura 4 – Representação de Aprendizado por Reforço

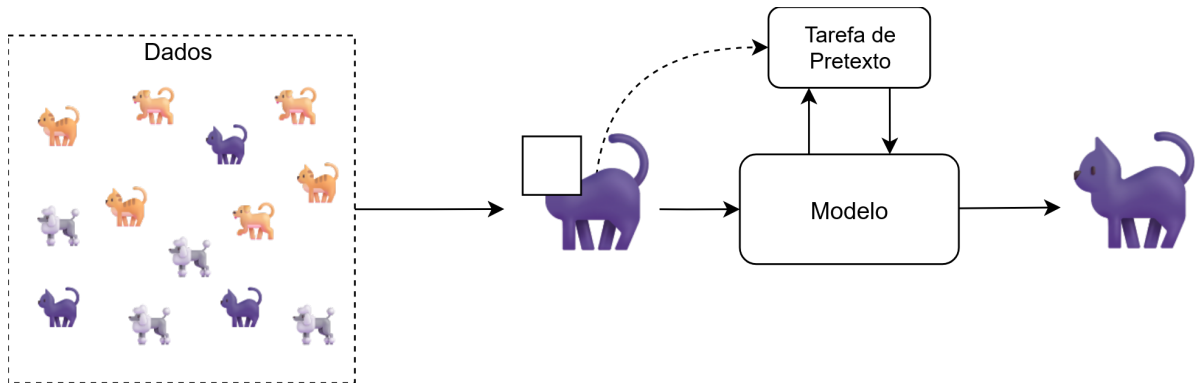


Fonte: o autor

Finalmente, um outro paradigma que têm ganhado expressividade e diversos avanços nos últimos anos é o aprendizado auto-supervisionado, que se afasta da divisão estrita entre dados rotulados e não rotulados. Nele, o próprio dado fornece o sinal de supervisão por meio de tarefas de pretexto, isto é, objetivos auxiliares cujos rótulos são gerados automaticamente a partir do conteúdo, sem qualquer anotação humana, como ocultar parte da entrada e treinar o modelo para reconstruí-la a partir do restante (Jaiswal; Babu et al., 2021). Resolvido o pretexto, as representações que o modelo aprendem deixam de ser específicas àquela tarefa e podem ser reaproveitada em problemas posteriores, como a

classificação ou detecção, muitas vezes sem novo treinamento (Jaiswal; Babu et al., 2021). A Figura 5 ilustra o processo.

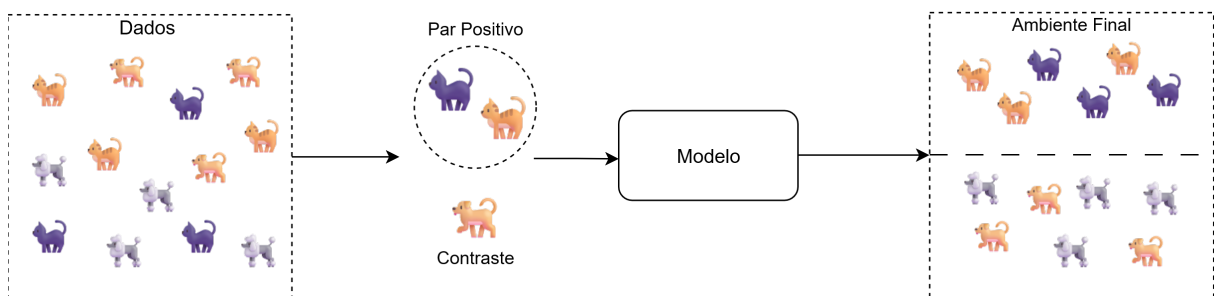
Figura 5 – Representação de Aprendizado Auto-Supervisionado



Fonte: o autor

Entre as abordagens auto-supervisionadas, destaca-se o aprendizado por contraste, que aprende não por reconstrução, como anteriormente exemplificado, mas por comparação. A ideia é aproximar as versões de uma mesma amostra em um espaço de representação – como por exemplo, dois recortes ou transformações distintas da mesma imagem – tratados como par positivo e, ao mesmo tempo, afastar as representações de amostras diferentes, tomadas como contrastes (Jaiswal; Babu et al., 2021). Dessa forma, o modelo é levado a descrever cada amostra pelo que a distingue das demais, estratégia que tem alcançado desempenho próximo ao do pré-treino supervisionado, ainda que sem o esforço de rotulagem (Jaiswal; Babu et al., 2021), exemplificado pela Figura 6. É o princípio de afastar o distinto e aproximar o semelhante que reaparece no módulo contrastivo do detector forense empregado neste trabalho.

Figura 6 – Representação de Aprendizado Contrastivo



Fonte: o autor

Embora os conceitos fundamentais de aprendizado de máquina tenham sido esta-

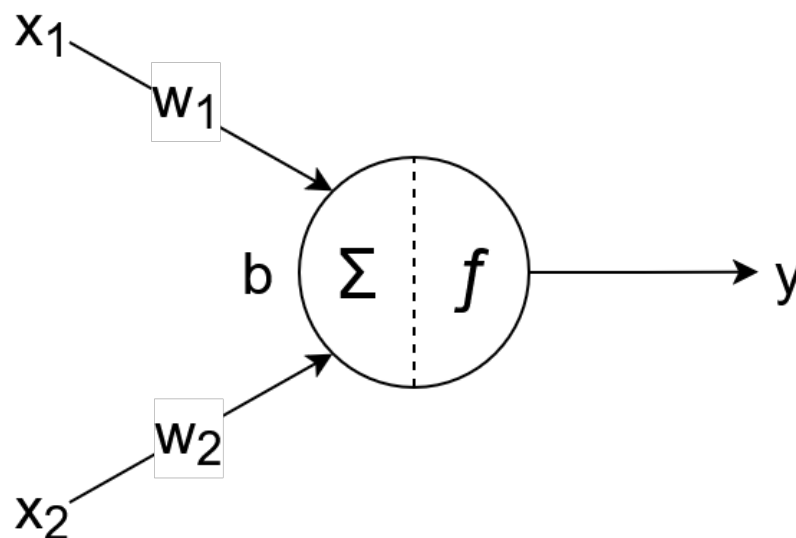
belecidos há quase um século, com contribuições embrionárias nas décadas de 1950 e 1960 (Dietterich, 2003), essa área de estudo tem recebido grande destaque nas últimas décadas. Esse ressurgimento deve-se principalmente ao aumento exponencial da capacidade computacional e à disponibilidade massiva de dados digitais. Também, a evolução do *hardware* – particularmente o advento de unidades de processamento gráfico de alto desempenho (GPU) – possibilitou o treinamento de modelos complexos, antes computacionalmente intratáveis, transformando o aprendizado de máquina em uma tecnologia fundamental para aplicações modernas em diversas áreas (Sarker, 2021).

2.3 REDES NEURAIS PROFUNDAS

Redes neurais profundas, comumente utilizadas nos paradigmas de aprendizados supervisionado e auto-supervisionado, são uma especialização de redes neurais artificiais. Diferentemente das técnicas tradicionais de *machine learning*, que requerem a engenharia manual de características, as abordagens baseadas em aprendizado profundo aprendem autonomamente representações complexas de dados brutos. Isso é possível devido à sua estrutura multicamada, que permite a extração progressiva de características de baixo nível – em uma imagem, por exemplo, bordas e linhas – até padrões de alto nível – como objetos e faces, no mesmo exemplo (Menghani, 2023).

A unidade mais básica de uma rede neural artificial é um neurônio artificial (referido neste estudo apenas como neurônio ou nó).

Figura 7 – Representação de um Neurônio Artificial



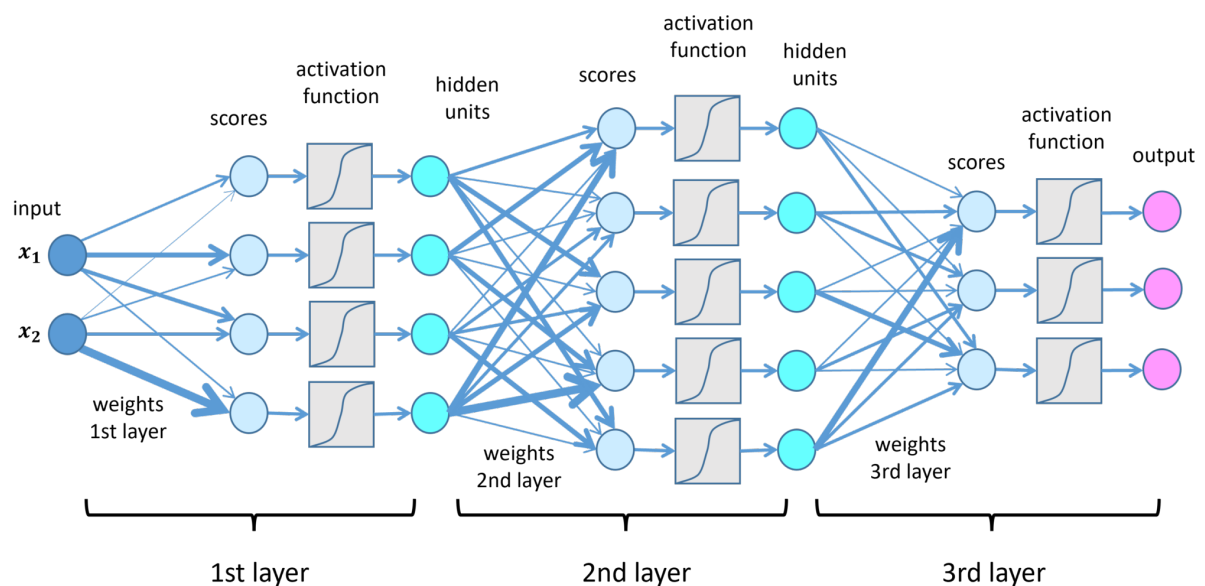
Fonte: o autor

No exemplo representado na Figura 7, um neurônio recebe dados de entrada, x_1 e

x_2 , e produz uma saída y . Para isso, cada entrada é multiplicada por seu respectivo peso, w_1 e w_2 , e somada junto a um termo de viés b – assim, a equação resultante é igual a $\Sigma = x_1w_1 + x_2w_2 + b$. Finalmente, aplica-se uma função de ativação f sobre a soma para converter esse valor em um intervalo desejado – a tangente hiperbólica produziria um número dentro do intervalo $(-1, 1)$, enquanto a sigmoide produziria um intervalo entre $(0, 1)$, por exemplo – resultando na saída y (Goodfellow; Bengio; Courville, 2016). Em outras palavras, os pesos indicam a importância, ou força, da conexão entre a entrada e o neurônio; o viés atua como um limiar de ativação que independe das entradas; e a função de ativação transforma uma entrada linear em uma saída não linear, o que permite um mapeamento complexo entre entradas e saídas.

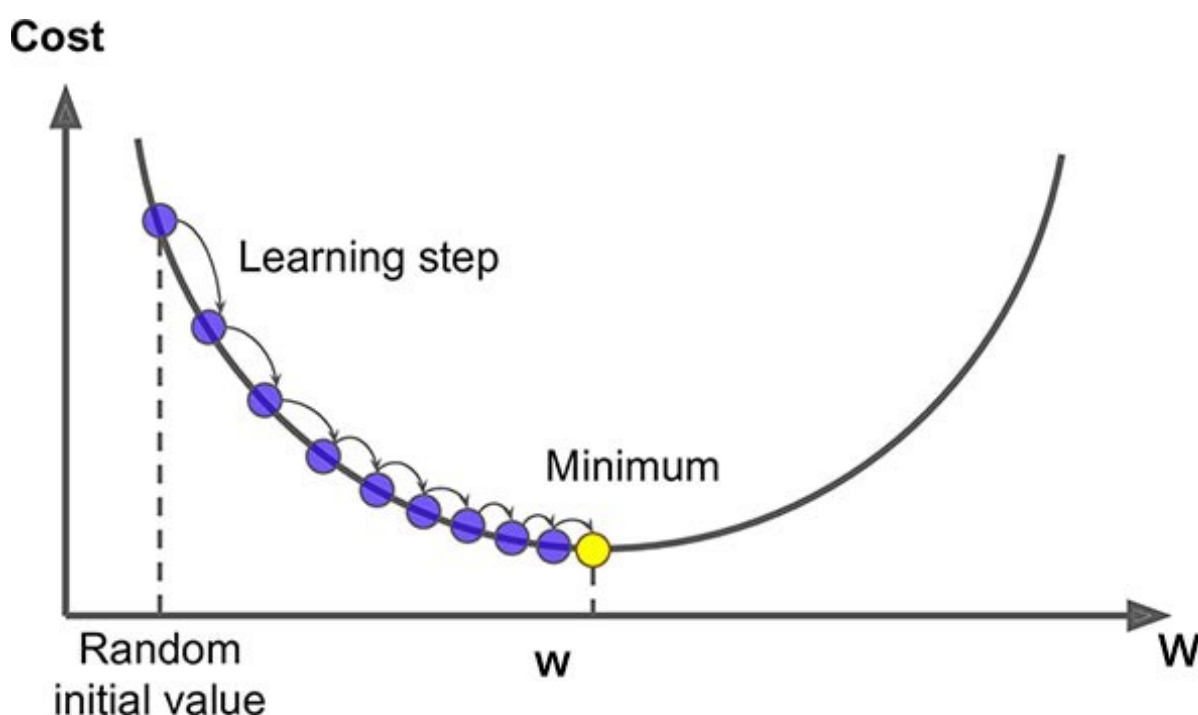
Uma rede neural artificial é formada pela interligação de neurônios, assim, uma camada da rede é denominada a partir de um grupo de nós interligados, que processam dados de uma maneira específica. Combinando uma camada de entrada, camadas intermediárias e uma camada de saída, obtém-se uma rede neural profunda, ilustrada na Figura 8. Dessa forma, cada camada recebe entradas ponderadas a partir das camadas anteriores, aplica uma função de ativação e propaga o resultado para as camadas subsequentes (Goodfellow; Bengio; Courville, 2016). Diferentes configurações destas, como a variação das conexões entre neurônios ou o emprego de funções de ativação distintas, têm por efeito especializações, ou habilidades de aprendizado específicas, assim, a utilização de múltiplas camadas permite a assimilação de representações hierárquicas complexas. (Alzubaidi et al., 2021).

Figura 8 – Representação de uma Rede Profunda



O processo de aprendizado da rede é denominado treinamento e consiste na estimação e ajuste dos parâmetros através de um algoritmo de retropropagação. Seu objetivo é calcular o gradiente de uma função que mede o erro entre o valor de saída computado e o esperado, além de ajustar os pesos e vieses dos neurônios na direção oposta ao gradiente, para minimizar o erro. Esse processo é executado em cada camada, propagando o erro desde a camada de saída até a de entrada, de forma iterativa, por múltiplas épocas, até que a rede convirja para uma solução otimizada (Goodfellow; Bengio; Courville, 2016), como ilustrado na Figura 9, em que o eixo y representa valores de erro e o eixo x valores de peso.

Figura 9 – Visualização do Gradiente Descendente



Fonte: <https://mlpills.dev/machine-learning/gradient-descent/>. Acesso em: 25 junho 2025

2.3.1 Redes Neurais Convolucionais

Redes neurais convolucionais são tipos de redes neurais profundas especialmente úteis na área de visão computacional, especializadas na computação de dados estruturados em topologia de malha, representados matricialmente. O que propicia essa propriedade é o emprego de operações de convolução em pelo menos um módulo da rede, o que consiste em uma mudança no processamento de entrada dos neurônios: ao invés da simples soma ponderada pelos pesos, um cálculo é efetuado a partir da aplicação de um filtro sobre um *input* (Goodfellow; Bengio; Courville, 2016). Dessa forma, a estrutura de um modelo básico

combina camadas convolucionais com camadas de subamostragem, todas esparsamente conectadas (Alzubaidi et al., 2021).

Convolução é uma operação matemática sobre duas funções para a criação de uma terceira, que representa, em termos simplórios, a sobreposição delas. No contexto de redes convolucionais, consiste na multiplicação de duas matrizes seguida de uma soma (Wang; Raj, 2017), como demonstrado na Figura 10.

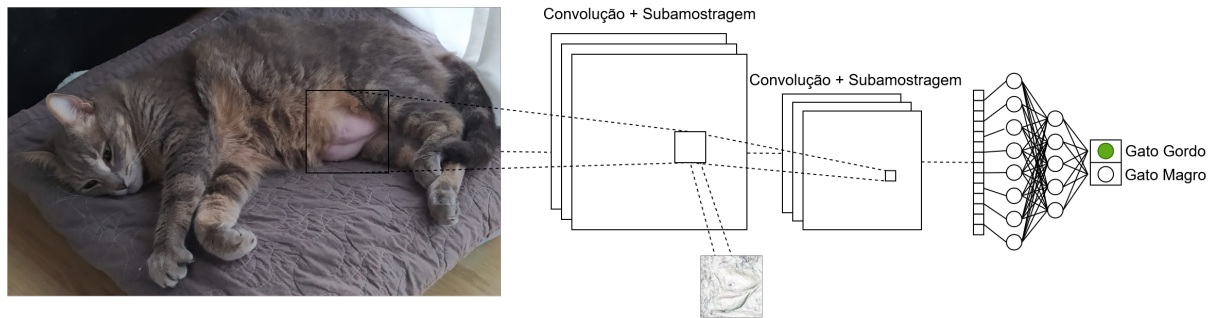
Figura 10 – Visualização de uma Operação de Convolução

$$\begin{bmatrix} 22 & 15 & 1 & 3 & 60 \\ 42 & 5 & 38 & 39 & 7 \\ 28 & 9 & 4 & 66 & 79 \\ 0 & 2 & 25 & 12 & 17 \\ 9 & 14 & 2 & 51 & 3 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 29 & 12 & 64 \\ 38 & 41 & 32 \\ 13 & 80 & 81 \end{bmatrix}$$

Fonte: (Wang; Raj, 2017)

Um filtro – ou *kernel*, que é uma matriz de parâmetros aprendíveis – desliza sobre os dados de entrada, a um passo definido, computando o produto escalar entre seus pesos e os valores correspondentes na região coberta, de forma a criar um mapa de características que representa a presença de padrões específicos detectados pelo filtro (Goodfellow; Bengio; Courville, 2016). Dessa forma, certas configurações, como o tamanho e os pesos do *kernel*, ou o número de passos de deslizamento, controlam a resolução espacial e a capacidade de modelar relações de vizinhança, como figurativamente representado na Figura 11, em que a escolha de um filtro inicial para a detecção de bordas permite o treinamento da rede para a distinção entre gatos magros e gordos. Assim, uma camada convolucional não se limita a um único filtro, mas aplica vários em paralelo, cada um sensível a um padrão diferente, de modo que produz um conjunto de mapas de características, um para cada filtro.

Figura 11 – Representação da Classificação de uma Imagem por uma Rede Convolucional



Fonte: o autor

Outro importante aspecto das redes convolucionais são as camadas de subamostragem, que reduzem a dimensão espacial dos dados de entrada ao passo que preservam suas características mais relevantes, processo conhecido como *pooling*. Atingem isso com a obtenção de apenas uma amostra para cada região analisada, para isso, empregam estratégias como *max-pooling* (extração do maior valor de entrada) e *average-pooling* (extração da média dos valores de entrada). A combinação de camadas convolucionais com camadas de subamostragem confere a essas redes três propriedades fundamentais: a invariância a pequenas transformações, distorções e translações da entrada, que permite que características sejam detectadas independentemente de sua localização; a capacidade de extrair hierarquias espaciais através da redução progressiva de dimensionalidade; e a redução do custo computacional das camadas subsequentes (Wang; Raj, 2017).

Dessa forma, a extração de *features* acontece conforme a entrada percorre a rede. Utilizando como exemplo uma imagem para a entrada, as camadas iniciais capturam informações de baixo nível, como bordas, cantos e texturas. Em camadas intermediárias, essas informações passam a compor padrões semânticos locais, como delimitações de objetos, até que, em camadas mais profundas, transformam-se em caracterizações globais de alto nível, como formas completas. Isso permite que as camadas finais possam prever, ou extrair, representações complexas e abstratas (Alzubaidi et al., 2021), como o exemplo utilizado anteriormente – a classificação entre gatos gordos e magros.

2.3.1.1 Reconhecimento Óptico de Caracteres (OCR)

O reconhecimento óptico de caracteres (OCR), no contexto de computação, descreve um tipo de software que tem por objetivo a conversão de texto (impressos ou presentes em imagens) a um formato digital, que possa ser processável por máquina (Islam; Islam; Noor, 2016). Originalmente, os métodos de OCR baseavam-se em técnicas de segmentação da visão computacional clássica junto a heurísticas, como a segmentação de linhas e caracteres por projeções de histograma, seguida pela aplicação de regras heurísticas para identificar cada símbolo (Shi; Bai; Yao, 2015). No entanto, as abordagens que utilizam o aprendizado

de máquina – em especial, redes convolucionais – têm se mostrado mais eficientes e precisas (Shi; Bai; Yao, 2015).

De acordo com Islam; Islam; Noor (2016), um processo de reconhecimento de caracteres moderno tipicamente passa pelo seguinte *pipeline*:

1. Aquisição da imagem: obtém-se a imagem com o texto a partir de uma fonte externa, como *scanner* ou câmera;
2. Pré-processamento: aplicam-se técnicas como remoção de ruído, operações morfológicas e limiarização para melhorar a qualidade da imagem;
3. Segmentação de caracteres: separam-se os caracteres através da análise de componentes conectados, perfis de projeção ou métodos avançados para tratar textos sobrepostos ou fragmentados;
4. Extração de *features* e classificação de caracteres: com os caracteres separados, utiliza-se uma rede convolucional para a extração e classificação de seus padrões;
5. Pós-processamento: para refinar os resultados e aumentar a acurácia, combinam-se técnicas de processamento de linguagem natural, como corretores ortográficos e dicionários com modelos probabilísticos, como cadeias de Markov e N-gramas.

Neste estudo, contudo, o OCR não cumpre o papel de extrair conteúdo para análise, mas apenas o de localizar as regiões que contêm texto em uma página. Como se detalha na metodologia, são essas coordenadas que orientam a geração de documentos falsificados, indicando quais campos do documento podem ser editados.

2.3.2 Transformadores

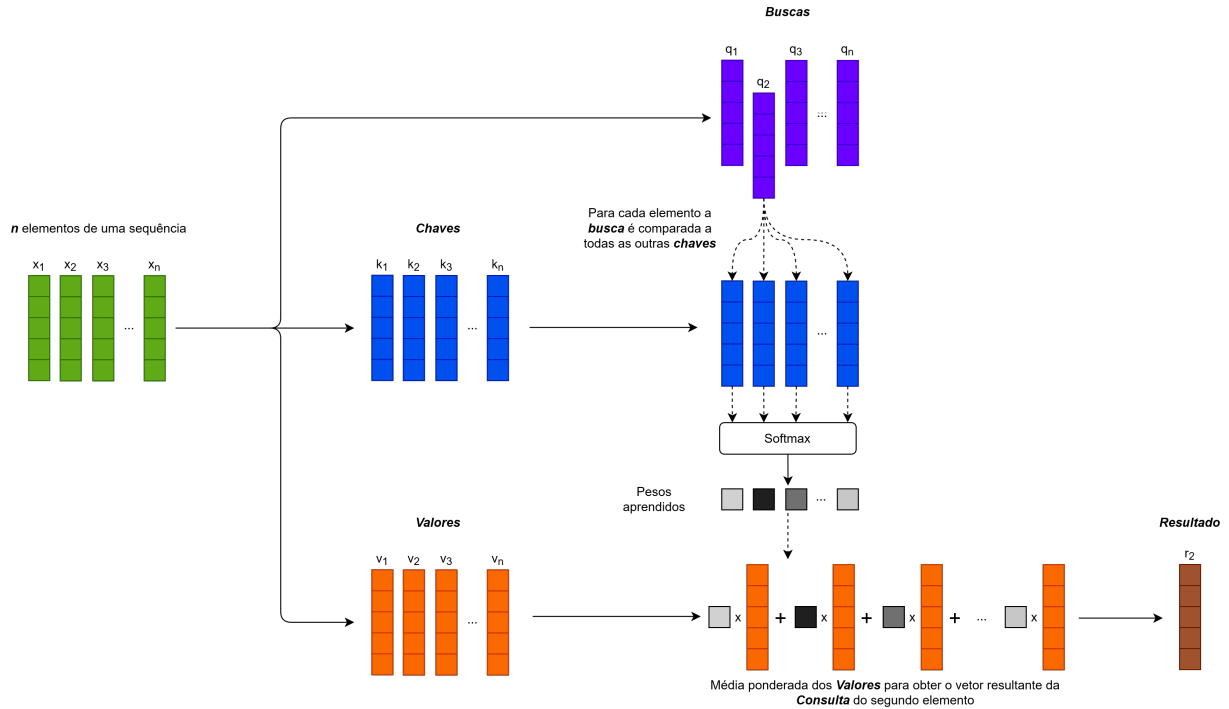
As redes neurais vistas até aqui operam sobre entradas de dimensão fixa e estrutura regular, como a grade de *pixels* de uma imagem. Muitos problemas, porém, envolvem sequências de comprimento variável, em que o significado de cada elemento, denominado *token*, depende de sua relação com os demais, como uma frase e suas palavras, por exemplo. Processar uma sequência de forma estritamente ordenada, um *token* de cada vez, é custoso e tende a enfraquecer as dependências entre posições afastadas, pois o sinal precisa atravessar todos os *tokens* intermediários (Vaswani et al., 2017). A arquitetura *transformer* surge com o intuito de superar essa limitação, pois abandona o processamento sequencial e passa a relacionar todos os elementos de uma só vez, apoiando-se inteiramente em um mecanismo de atenção (Vaswani et al., 2017).

Esse mecanismo de atenção tem por objetivo emular um dicionário de chaves e valores através da produção, para cada *token* da sequência, de uma nova representação que reúna a informação dos demais *tokens* relevantes a ele. Para isso, é necessário primeiro decidir quais dos outros *tokens* são relevantes e, em seguida, recolher deles a informação a

incorporar. Assim, a atenção cumpre as duas tarefas atribuindo a cada *token* três vetores, obtidos por transformações lineares aprendidas: o primeiro, denominado consulta (*query*), expressa o que aquele elemento procura nos demais; o segundo, a chave (*key*), o que cada elemento oferece; e o terceiro, o valor (*value*), que representa a informação. Dessa forma, a consulta de um *token* é comparada com a chave de cada um dos demais por meio do produto escalar, operação que produz um escore maior à medida que os vetores forem mais semelhantes, e que se traduz na relevância entre os elementos. Essa operação resulta na recuperação do valor, conforme relevância de sua chave correspondente (Vaswani et al., 2017).

Esses escores de relevância, porém, são números de magnitude arbitrária, assim, o conjunto de escores que um *token* atribui aos demais passa por uma função *softmax*, que converte vetores em uma distribuição de probabilidades, isto é, converte os escores em pesos positivos cuja soma é igual a um (Vaswani et al., 2017). Dessa forma, cada peso exprime a fração de atenção que o *token* dedica a cada um dos outros. A nova representação do elemento é então obtida pela combinação dos valores de todos os demais, cada um multiplicado pelo seu respectivo peso. Como os pesos maiores recaem sobre os *tokens* mais relevantes, são as informações destes que predominam na representação resultante, ao passo que as dos menos pertinentes quase não a influenciam (Vaswani et al., 2017). Esse procedimento é executado de uma só vez para todos os *tokens* da sequência. A Figura 12 descreve de forma simplificada o mecanismo de atenção e pode ajudar na compreensão de todo o fluxo de operações.

Figura 12 – Representação do Mecanismo de Atenção



Fonte: adaptação de <https://epichka.com/blog/2023/qkv-transformer/>. Acesso em: 13 junho 2026

Para enriquecer ainda mais essa descrição, o mecanismo de atenção é replicado em paralelo em várias versões independentes, chamadas cabeças de atenção. Cada cabeça opera sobre projeções distintas das consultas, chaves e valores, o que lhe permite observar e aprender um tipo diferente de relação entre os *tokens* (Vaswani et al., 2017). As representações produzidas pelas várias cabeças são, ao final, concatenadas e combinadas em uma só, reunindo as diferentes perspectivas em uma única descrição de cada elemento. Esse bloco de atenção multi-cabeça constitui a parte central de uma camada do codificador, que o combina ainda a uma pequena rede neural aplicada a cada *token* individualmente (Vaswani et al., 2017). Assim, o codificador completo é formado pelo empilhamento de várias dessas camadas idênticas, de modo que a representação de cada elemento é refinada sucessivamente, camada após camada. No entanto, por construção, o mecanismo de atenção ignora a ordem dos *tokens*. Para reintroduzi-la, cada elemento recebe uma codificação que registra sua posição na sequência, somada à sua representação antes da primeira camada (Vaswani et al., 2017).

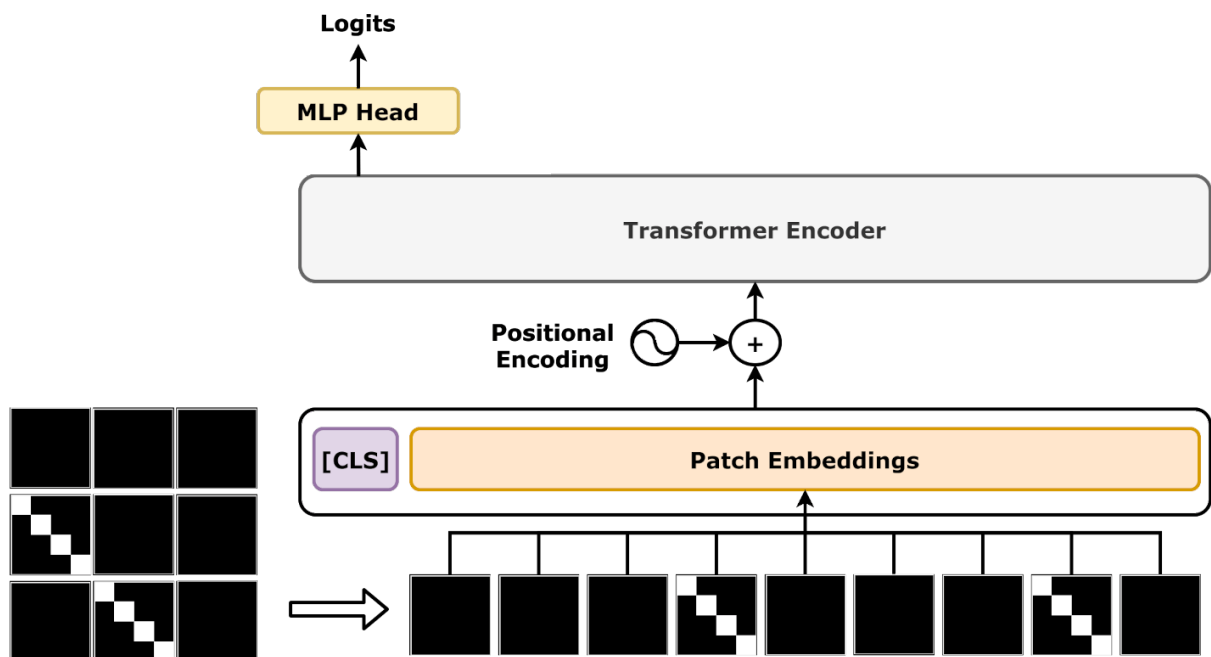
Ao final de todas as camadas, para cada *token*, o codificador produz uma representação sensível ao contexto, que resume sua relação com todo o restante da sequência (Vaswani et al., 2017). Essa representação pode ser utilizada diretamente, como descrição da entrada, ou entregue a um decodificador que a converte em uma saída desejada. É essa ideia de produzir um descritor da entrada que os *vision transformers* levam para o domínio das imagens (Dosovitskiy et al., 2021).

2.3.2.1 Vision Transformers

Os *vision transformers* levam o mecanismo de atenção, originalmente concebido para texto, ao domínio das imagens. Uma vez que o codificador anteriormente descrito opera sobre uma sequência de vetores, é preciso converter a imagem a esse formato. Para isso, ela é dividida em fragmentos quadrados de tamanho fixo, denominados *patches*, e cada um é achatado e projetado linearmente em um vetor, chamado *patch embedding*, de modo que o conjunto de fragmentos passa a formar uma sequência de *tokens* (Dosovitskiy et al., 2021).

Essa sequência é então entregue ao codificador *transformer*, cuja atenção relaciona as diferentes regiões da imagem e descreve cada fragmento no contexto dos demais, resultando em uma representação para cada *token* (Dosovitskiy et al., 2021). Como, porém, muitas tarefas exigem um único descritor da imagem inteira, acrescenta-se à sequência, antes do processamento, um *token* aprendido, conhecido como *token* de classe (CLS). Por meio da atenção, o CLS reúne a informação de todos os fragmentos, de modo que o vetor por ele produzido na saída do codificador serve como representação da imagem inteira (Dosovitskiy et al., 2021). Esse processo é representado na Figura 13, que ilustra ainda a etapa final de uma tarefa de classificação, na qual a representação do CLS passa por uma camada classificadora (*MLP head*) e resulta nos *logits*, as pontuações atribuídas a cada classe possível.

Figura 13 – Representação de um Vision Transformer



Fonte: https://en.wikipedia.org/wiki/Vision_transformer. Acesso em: 14 junho 2026

Essa forma de processar a imagem tem, contudo, uma contrapartida. Ao tratá-la

como uma sequência genérica de fragmentos, relacionados apenas por atenção, os *vision transformers* não contam com as suposições que as redes convolucionais incluem em sua estrutura, como a localidade e a invariância a translações. Privados dessas hipóteses de partida, precisam aprendê-las diretamente dos dados, o que exige um pré-treino em larga escala (Dosovitskiy et al., 2021). Porém, cumprida essa condição, igualam ou superam as redes convolucionais, muitas vezes a um custo computacional menor (Dosovitskiy et al., 2021). É sobre descritores visuais desse tipo que se apoia a extração de características empregada neste trabalho.

2.3.3 Representação por Vetores de Características

Os descritores visuais produzidos por arquiteturas como os *vision transformers*, vistos anteriormente, são um caso particular da ideia mais geral de representar um dado por meio de um vetor de características, ou *embeddings*. Como aponta Bengio; Courville; Vincent (2014), o desempenho de um sistema de aprendizado de máquina depende fortemente da representação sobre a qual ele opera, antes obtida por um processo manual e trabalhoso de engenharia de características, e hoje aprendida por uma rede neural, que infere do dado bruto quais deve extrair. Essa representação condensa uma entrada, como uma imagem ou um texto, em um vetor numérico de dimensão fixa, no qual cada componente exprime um aspecto aprendido do conteúdo. O que a torna útil é a organização do espaço em que esse vetor reside, construído de modo que entradas semanticamente semelhantes ocupem posições próximas e entradas distintas fiquem afastadas. Dessa forma, a proximidade entre dois vetores passa a refletir a similaridade de conteúdo entre as entradas que os geraram, e uma imagem complexa passa a dispor de uma espécie de impressão digital numérica, diretamente comparável à de outras por meio de uma medida de distância (Bengio; Courville; Vincent, 2014).

Uma forma de medir essa proximidade é a similaridade de cosseno, que compara dois *embeddings* pelo ângulo formado entre eles, sem considerar suas magnitudes. Quanto menor esse ângulo, mais alinhados os vetores e mais semelhantes as entradas que representam, de modo que o que importa é a direção de cada vetor, e não o seu comprimento.

Esses vetores, no entanto, costumam residir em espaços de alta dimensionalidade, o que demanda técnicas de redução para evitar problemas relacionados à *maldição da dimensionalidade*, termo cunhado por Bellman (1957), que explica que, ao lidar com dados em espaços de alta dimensão, sua amostragem e modelagem tornam-se progressivamente mais esparsas e difíceis, dado que o volume do espaço cresce exponencialmente.

Entre as técnicas de redução de dimensionalidade, destaca-se a análise de componentes principais (*Principal Component Analysis*, PCA). De acordo com Jolliffe; Cadima (2016), o método reduz a dimensionalidade de um conjunto de dados ao construir novas variáveis não correlacionadas entre si, chamadas componentes principais, que sucessivamente maximizam a variância dos dados. Como os primeiros componentes concentram a

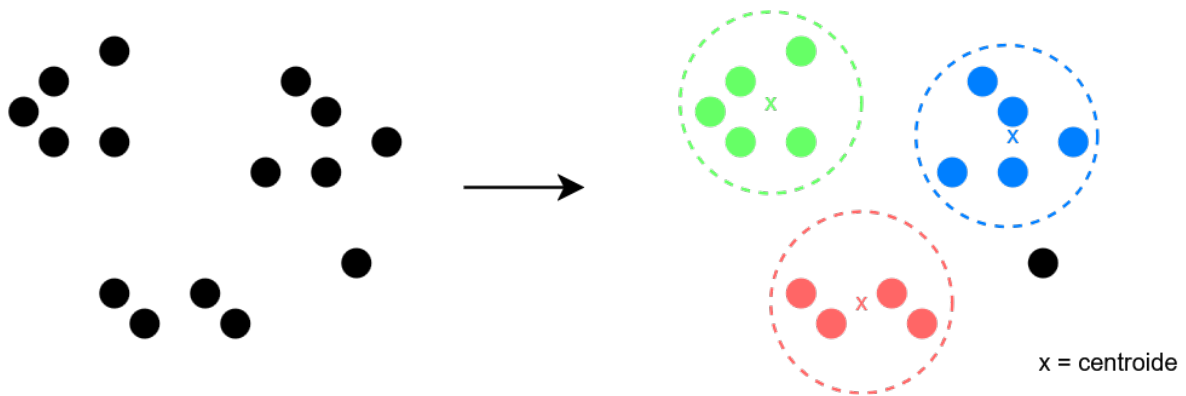
maior parte dessa variância, projetar os vetores apenas sobre elas descarta as direções de menor variação e produz uma representação de dimensão reduzida que retém quase toda a informação original (Jolliffe; Cadima, 2016). A esse procedimento pode ser acrescentada uma etapa de normalização conhecida como *whitening*, que reescala os componentes para que todas passem a ter a mesma variância, evitando que dimensões de maior magnitude dominem a comparação entre os vetores.

2.4 ALGORITMOS DE AGRUPAMENTO

No contexto de aprendizado de máquina, as técnicas de agrupamento, ou *clustering*, têm por objetivo a organização de uma coleção de dados em grupos, ou *clusters*, de forma que os elementos dentro de um mesmo grupo tenham muito mais similaridades entre si do que entre elementos em outros grupos. Essas técnicas geralmente são utilizadas quando não se tem conhecimento sobre como os dados podem ser categorizados, dessa forma, tradicionalmente são associadas à aprendizagem não supervisionada (Grira; Crucianu; Boujemaa, 2005). Assim, têm como propósito descobrir estruturas, relações, associações ou hierarquias ocultas dentro da coleção, de maneira a proporcionar melhor compreensão sobre os dados e seus processos subjacentes de criação (Fullér; Károly; Galambos, 2018).

Como trata-se de uma larga área de estudo, os algoritmos de *clustering* variam significativamente em sua compreensão do que constitui um *cluster* e de como este pode ser encontrado de forma eficiente, dessa forma, noções populares incluem grupos com pequenas distâncias entre membros, áreas densas do espaço de dados, intervalos ou distribuições estatísticas específicas (Fullér; Károly; Galambos, 2018). Por fins de brevidade, abordam-se aqui três abordagens representativas, baseadas em distância, em densidade e em hierarquia.

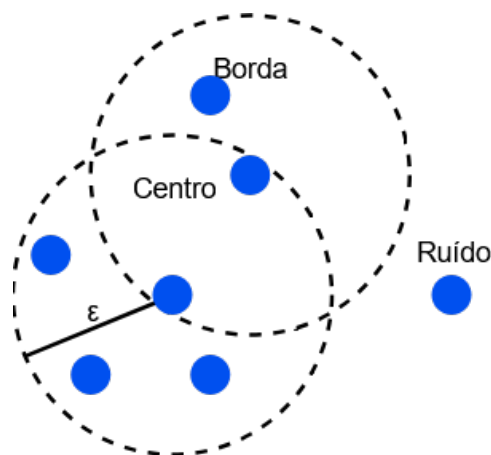
Para os algoritmos baseados em distância, cada *cluster* é representado por um protótipo e a atribuição de elementos a estes é guiada pela minimização de alguma medida de dissimilaridade. Em termos mais concretos, tanto os *clusters* quanto os dados são tipicamente traduzidos como pontos em um espaço. Busca-se então a minimização de uma função de custo, como a soma das distâncias quadráticas entre cada elemento e um *cluster*. Isso acontece através do cálculo iterativo da função de custo seguido do ajuste posicional do *cluster*, processo repetido alternadamente até que o valor de custo fique abaixo de um limiar determinado, garantindo a convergência do algoritmo, ou seja, a posição ótima do agrupamento (Grira; Crucianu; Boujemaa, 2005; Fullér; Károly; Galambos, 2018).

Figura 14 – Representação de um Algoritmo de *Clustering* por Distância

Fonte: o autor

Os algoritmos baseados em densidade consideram que *clusters* são regiões de alta densidade de dados separadas por áreas de baixa densidade, assim, tanto o formato dos agrupamentos quanto a distribuição dos elementos podem ser feitos de forma arbitrária. O algoritmo DBSCAN pode ser utilizado para exemplificar esse conceito: definem-se os parâmetros ϵ (raio de vizinhança) e *MinPts* (número mínimo de pontos ou elementos), assim, um ponto considera outro como vizinho quando está dentro de ϵ . Então, conta-se o número de vizinhos para cada ponto: aqueles com vizinhança densa – isto é, número de vizinhos maior que *MinPts* – tornam-se pontos centrais e formam núcleos de *clusters*; aqueles atingíveis por vizinhança dentro de um *cluster* são absorvidos e chamados de pontos de fronteira; e aqueles pouco densos e não atingíveis são classificados como ruído. (Grira; Crucianu; Boujemaa, 2005).

Figura 15 – Representação do Algoritmo DBSCAN



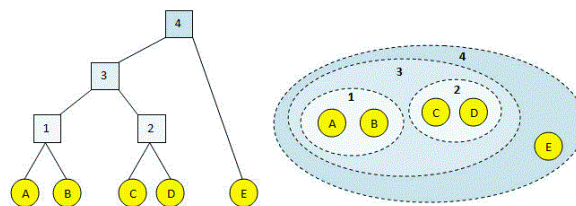
Fonte: o autor

Os algoritmos hierárquicos, por sua vez, não produzem uma divisão única, mas uma

hierarquia de grupos aninhados, organizados conforme usa semelhança (Fullér; Károly; Galambos, 2018). Na variante aglomerativa, a mais comum, cada elemento começa como um grupo isolado, e a cada passo os dois grupos mais semelhantes são fundidos em um só, repetindo-se a operação até restar um único grupo que contém toda a coleção (Fullér; Károly; Galambos, 2018). A distância entre dois elementos é dada por uma medida como a distância de cosseno, ao passo que, como um grupo pode reunir vários elementos, um critério de ligação (denominado *linkage*) estende essa distância aos grupos, tomando a distância entre dois deles como a menor distância entre seus elementos (*linkage* simples), a maior delas (*linkage* completo), ou a média entre todas (Fullér; Károly; Galambos, 2018).

Essa sucessão de fusões forma uma estrutura em árvore, em que cada nível corresponde a um agrupamento de granularidade distinta (Grira; Crucianu; Boujemaa, 2005). Assim, não é preciso fixar de antemão o número de grupos, que decorre do ponto em que a árvore é cortada, em geral por um limiar de distância abaixo do qual dois grupos são tidos como próximos o suficiente para serem unidos (Fullér; Károly; Galambos, 2018).

Figura 16 – Representação do Agrupamento Aglomerativo



Fonte: <https://medium.com/@khushivirpariya/agglomerative-clustering-with-graph-557668ed1f2e>.
Acesso em: 14 junho 2026.

2.4.1 Detecção de Anomalias

A detecção de anomalias é definida como o processo de identificar padrões em um conjunto de dados cujo comportamento difere significativamente do esperado, assim, pode ser tratado como um problema de identificação de *outliers* ou ruídos em algoritmos de agrupamento. Quando identificadas, embora nem sempre representem atividades maliciosas, frequentemente indicam eventos de interesse que merecem investigação (Agrawal; Agrawal, 2015; Fullér; Károly; Galambos, 2018).

A metodologia típica segue três etapas principais: parametrização e pré-processamento, onde os dados são normalizados em formatos comuns; treinamento, onde são construídos modelos de comportamento normal; e finalmente, detecção, em que uma nova observação é comparada aos modelos construídos (Agrawal; Agrawal, 2015). Dessa forma, a detecção de anomalias baseada em *clustering* funciona estabelecendo modelos de normalidade através do agrupamento de dados e, posteriormente identificando novos elementos que apresentam desvios expressivos desses modelos de referência.

3 TRABALHOS CORRELATOS

As subseções seguintes apresentam os mais recentes trabalhos com temática ou abordagem semelhante.

3.1 VISÃO GERAL

A verificação automatizada da autenticidade de documentos é um problema amplamente investigado no campo da forense digital (Zhao et al., 2026), ainda que a bibliografia que aborde especificamente sobre o contexto de certificados acadêmicos seja escassa. Durante a pesquisa pelo estado da arte, observou-se que os trabalhos correlatos atacam o problema por duas frentes complementares: a prevenção da adulteração, por meio de garantias criptográficas atreladas ao momento da emissão; e a sua detecção, por meio da análise de artefatos do próprio documento.

Na frente da prevenção, encontram-se propostas como a de Boonkrong (2024), que emprega funções criptográficas para detectar modificações em documentos previamente submetidos, onde armazenam-se os valores de *hash* dos arquivos originais e legítimos, de forma que validações posteriores possam ser comparadas com o certificado primário. Contudo, essas abordagens pressupõem que o documento já tenha sido registrado por uma fonte confiável, garantindo sua integridade apenas a partir desse instante, ou seja, nada informam sobre a legitimidade de um documento apresentado pela primeira vez, no momento do próprio registro – precisamente a brecha sobre a qual este trabalho atua.

No que tange à detecção, por sua vez, é predominante o emprego de estratégias de visão computacional, como no artigo de Jaiswal; Sharma; Yadav (2022), que utiliza *autoencoders* convolucionais para a extração de características em imagens hiperespectrais, focando em identificar incompatibilidades entre tintas. A análise de imagens é frequentemente combinada com técnicas complementares para melhorar a robustez da detecção: Alameri et al. (2023) propõem uma abordagem não supervisionada que utiliza correlações entre os espectros dos materiais dos documentos para gerar redes ponderadas, aplicando algoritmos de *clustering* para identificar inconsistências entre eles; James; Gupta; Raviv (2020) introduziram outra perspectiva ao reformular o problema como comparação de grafos, obtendo, via OCR, caixas delimitadoras de tamanho entre caracteres e utilizando-as para o treinamento de classificadores que detectam a manipulação de *pixels*. Em sequência, seguem os trabalhos que guiaram a concepção da estratégia adotada na presente pesquisa.

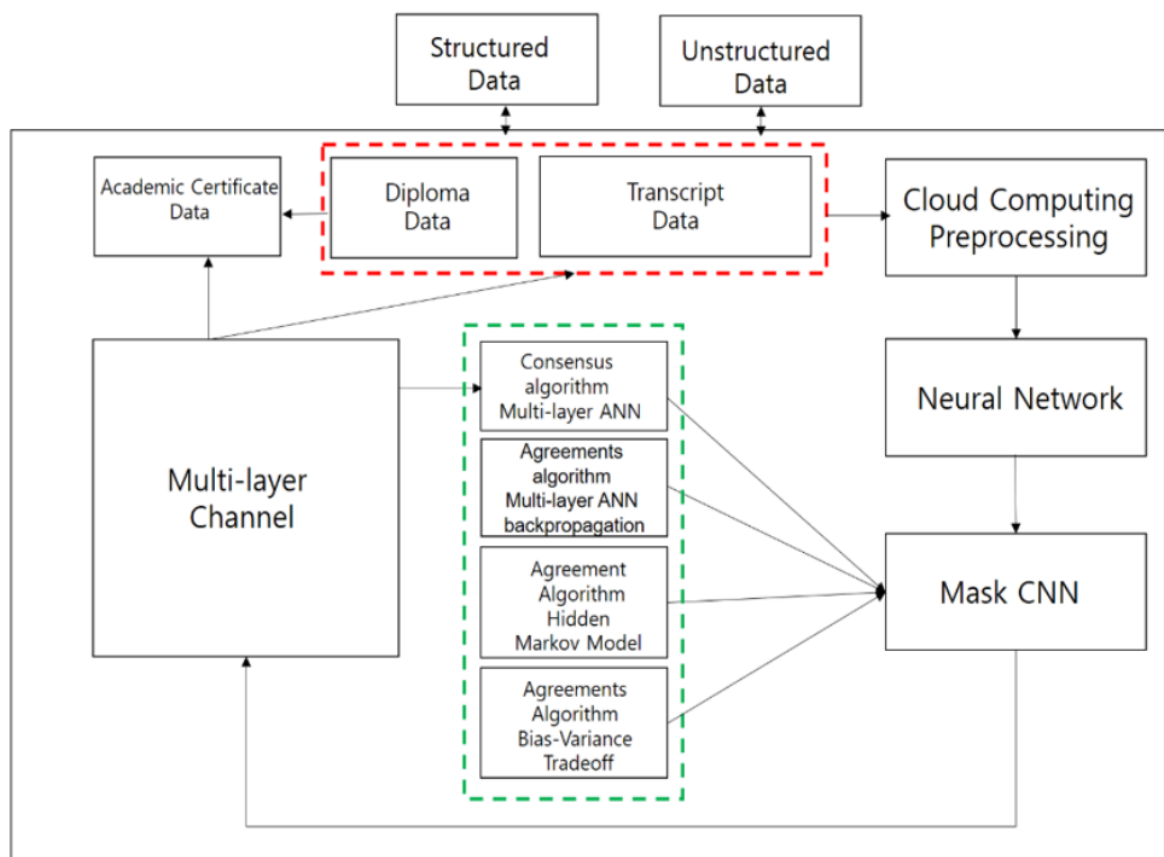
3.2 INTEGRANDO APRENDIZADO DE MÁQUINA À *BLOCKCHAIN* PARA PREVENÇÃO E DETECÇÃO DE FRAUDES EM DIPLOMAS

Para o problema de prevenção de fraudes de diplomas, o artigo de Kim (2022) propõe uma *blockchain* que incorpora algoritmos de aprendizado de máquina em diversas

partes do processo de verificação de documentos e de consenso da rede. Além disso, de forma semelhante ao trabalho de Boonkrong (2024), quando um certificado é aceito, seu *hash* é calculado e integrado ao seu registro na rede, permitindo sua verificabilidade, de forma que qualquer adulteração seja facilmente detectada.

O autor apresenta o fluxo para submissão de um diploma na *blockchain* em quatro etapas principais, conforme Figura 17.

Figura 17 – Representação da Arquitetura de Validação de Diplomas por Kim



Fonte: (Kim, 2022)

A primeira etapa, "*Cloud Computing Preprocessing*", consiste na entrada e pré-processamento do documento digitalizado e, através de um serviço em nuvem, normaliza os dados brutos da imagem, extrai metadados – como resolução, dimensões e contraste –, aplica correções de perspectiva e cria uma versão virtualizada semi-persistente do arquivo, preparada para ser processada pelas etapas consequentes, rejeitando entradas com formato e qualidade inconsistentes.

Em seguida, as imagens passam pela etapa "*Neural Network*", que utiliza *Faster R-CNN* para rapidamente filtrar e detectar artefatos visuais suspeitos. Através de sua rede de propostas regionais (*Region Proposal Network*), o algoritmo captura regiões de

interesse – como selos da universidade, assinaturas, marcas d'água e outros padrões – e cria escores de confiança para cada uma, identificando áreas possivelmente adulteradas.

Os documentos passam então para a etapa "*Mask CNN*", que através de um modelo *Mask R-CNN* e das regiões de interesse previamente identificadas, segmenta a imagem, criando máscaras binárias a nível de *pixel*, ou seja, para cada região, o algoritmo estima quais *pixels* pertencem ao documento legítimo e quais foram forjados. Essas máscaras são então encapsuladas como prova imutável dentro do bloco que será registrado na *blockchain*, além de servir como outra medida de escore de confiança do documento.

Por fim, ambas as pontuações de confiança são combinadas e passadas ao mecanismo de consenso dessa *blockchain*, representado pela etapa "*Multi-layer Channel*" que, ao invés de utilizar estratégias tradicionais como prova de trabalho ou prova de participação, combina múltiplos algoritmos de aprendizado de máquina. De forma geral, essa arquitetura é composta por quatro componentes principais:

- Um algoritmo baseado em uma rede neural multicamadas, que processa os escores de confiança fornecidos pelos processamentos anteriores e, a partir de certo limiar de confiança, imediatamente aprova o documento;
- Um algoritmo de aprendizado complementar à rede neural multicamadas, que cruza referências com padrões aprendidos de decisões anteriores – por exemplo, quando um diploma inicialmente validado como autêntico posteriormente se prova fraudulento – e ajusta os pesos da rede;
- Um algoritmo que utiliza um modelo oculto de Markov para, a partir de uma cadeia de regras, examinar padrões temporais e fornecer avaliações probabilísticas da autenticidade do documento;
- Um algoritmo que equilibra a complexidade (*overfitting*) com a generalização (*underfitting*) do modelo de rede neural, ou seja, procura encontrar *trade-off* ótimo entre viés e variância para decisões de consenso confiáveis.

Assim, cada nó da *blockchain* executa esses algoritmos e vota, com base na ponderação dos resultados, se devem incluir ou rejeitar o bloco com o diploma. Para qualquer decisão, exige-se quórum de pelo menos 2/3 de votos. Caso não seja atingido, seja por discordâncias das máscaras ou pela recusa de validadores de alta reputação, uma nova rodada de votação é iniciada com a reexecução das etapas "*Mask CNN*" e "*Multi-layer Channel*" com parâmetros ajustados. Esse ciclo se repete até obter consenso ou direcionar o diploma a uma auditoria humana. Dessa forma, quando um documento é aprovado na rede, é classificado como autêntico no *ledger*; quando reprovado, é sinalizado como fraudulento.

Por fim, quando um terceiro – como empresa, universidade ou empregador – deseja verificar a validade de um diploma já registrado, basta a verificação do *hash* do documento já submetido.

3.3 APLICANDO ANÁLISE MULTIMODAL A DOCUMENTOS

Diferentemente dos demais trabalhos discutidos neste capítulo, o de Jain; Wigington (2019) não trata da identificação de documentos falsificados, mas do problema geral de classificação de imagens de documentos. É aqui incluído não como trabalho correlato, e sim como inspiração metodológica, por demonstrar a eficácia de combinar características complementares na análise de documentos, ideia que norteou a concepção inicial deste trabalho.

O *paper* propõe uma abordagem multimodal para a classificação de imagens de documentos diversos em dezesseis categorias utilizando o *dataset* RVL-CDIP. A proposta combina a fusão de características visuais e textuais para a rotulagem entre imagens e classes. Para isso, segue o *pipeline*:

1. Pré-processamento: normaliza e redimensiona as imagens para utilizações posteriores;
2. Extração de texto: utiliza OCR para extrair texto das páginas;
3. Extração multimodal, em paralelo:

Modalidade textual: utiliza um modelos de linguagem para capturar informações semânticas dos textos extraídos;

Modalidade visual: extrai características das imagens a partir de uma rede convolucional;

4. Fusão multimodal: combina as extrações textuais e visuais;
5. Classificação final: utiliza uma rede convolucional, que tem como entrada a fusão multimodal, para classificar os documentos.

Para a modalidade textual, os autores trazem à tona o problema de que texto extraído por OCR pode ser muito ruidoso, contendo erros a nível de caracteres ou até palavras. Por isso, capturam representações do conteúdo em três diferentes granularidades.

A nível de sequência, empregam ULMFiT (*Universal Language Model Fine-tuning*), um modelo baseado em Redes Neurais Recorrentes – cuja explicação sobre o funcionamento foge do escopo do presente trabalho – que processa o documento como uma sequência de palavras, produzindo uma representação vetorial que preserva dependências de longo prazo e o contexto semântico entre termos distantes.

A nível de palavra, para representar cada uma, empregam FastText *embeddings*, um algoritmo de *embedding* de termos, de forma que palavras com significados similares fiquem próximas no espaço matemático. O vetor final do documento é calculado como a média dos *embeddings* de todas as palavras presentes.

A nível de caractere, para capturar padrões ortográficos, aplicam N-gramas de caracteres – sequências contínuas de n caracteres abduzidos de uma palavra – e criam um vetor numérico normalizado das ocorrências dos padrões obtidos.

Em resumo, as características de sequência preservam o contexto semântico geral do documento, as representações de palavra mantêm similaridades semânticas locais, e os N-gramas de caracteres oferecem robustez contra erros de OCR e palavras desconhecidas. Essas três representações são combinadas através de um método *ensemble*, que produz um vetor unificado que comporta essas *features* textuais.

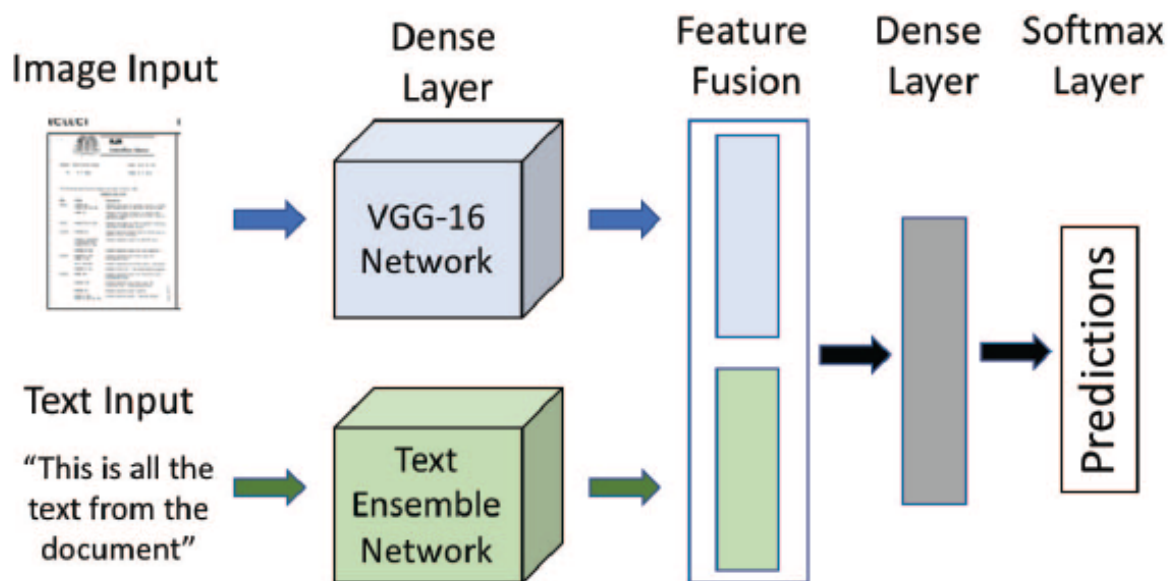
Para a modalidade visual, o trabalho emprega a arquitetura de rede VGG-16 (*Visual Geometry Group*) para extrair características hierárquicas através de suas camadas convolucionais, capturando padrões de layout, tipografia, elementos gráficos e padrões de formatação presentes nos documentos. Como saída, a rede produz um vetor multidimensional que representa uma codificação densa e compacta de todas as informações visuais relevantes do documento.

O principal interesse desse trabalho é a fusão das informações textuais e visuais, que tem por objetivo criar uma representação unificada, que preserve e potencialize as informações complementares de ambas as modalidades, e que permita que um modelo de classificação final explore sinergias entre essas diferentes características. Os autores propõem duas estratégias principais para combinar essas representações, das quais destaca-se a segunda, que combina os vetores anteriormente extraídos.

Essa abordagem explora quatro métodos distintos de junção. De forma geral, o primeiro é a concatenação simples, onde os vetores de características textuais e visuais são diretamente concatenados para formar um vetor unificado. O segundo método utiliza adição elemento a elemento, somando diretamente as representações de ambas as modalidades. O terceiro emprega *compact bilinear pooling*, uma técnica mais sofisticada que calcula o produto externo entre os vetores de características para capturar interações complexas entre as modalidades, permitindo que o modelo de classificação posterior aprenda correlações não-lineares entre informações visuais e textuais. Por fim, o quarto método implementa *multimodal gated units*, que utilizam mecanismos de atenção para aprender uma função de controle que determina automaticamente como ponderar e combinar as características de cada modalidade, o que permite que o modelo posterior adapte dinamicamente a importância relativa de informações visuais ou textuais dependendo do contexto específico do documento – para documentos altamente textuais como contratos ou relatórios científicos, as características semânticas podem ser mais discriminativas, enquanto para documentos com layouts visuais distintivos como formulários ou apresentações, as características visuais podem ser mais relevantes.

Em sequência, para realizar a classificação final, os autores utilizam uma camada densa e uma camada final *softmax* para a predição, como ilustra a Figura 18.

Figura 18 – Representação da Arquitetura de Fusão Multimodal por Jain; Wigington



Fonte: (Jain; Wigington, 2019)

Finalmente, o artigo compartilha as conclusões finais, onde explicita como essa abordagem superou consistentemente os métodos que utilizam apenas uma modalidade, com o método de adição elemento a elemento curiosamente alcançando os melhores resultados para a fusão multimodal, atingindo uma acurácia de 93,6% no *dataset* RVL-CDIP.

3.4 TRATANDO A AUTENTICAÇÃO DE DOCUMENTOS ACADÊMICOS COMO UM PROBLEMA DE AGRUPAMENTO

O trabalho de Mohammed; Nwobodo; Ekene (2024) lida diretamente com o problema de verificação da autenticidade de certificados acadêmicos. Para isso, propõe uma abordagem baseada em *clustering*, fundamentado na premissa de que documentos legítimos apresentam padrões consistentes de características que podem ser identificados através desses agrupamentos.

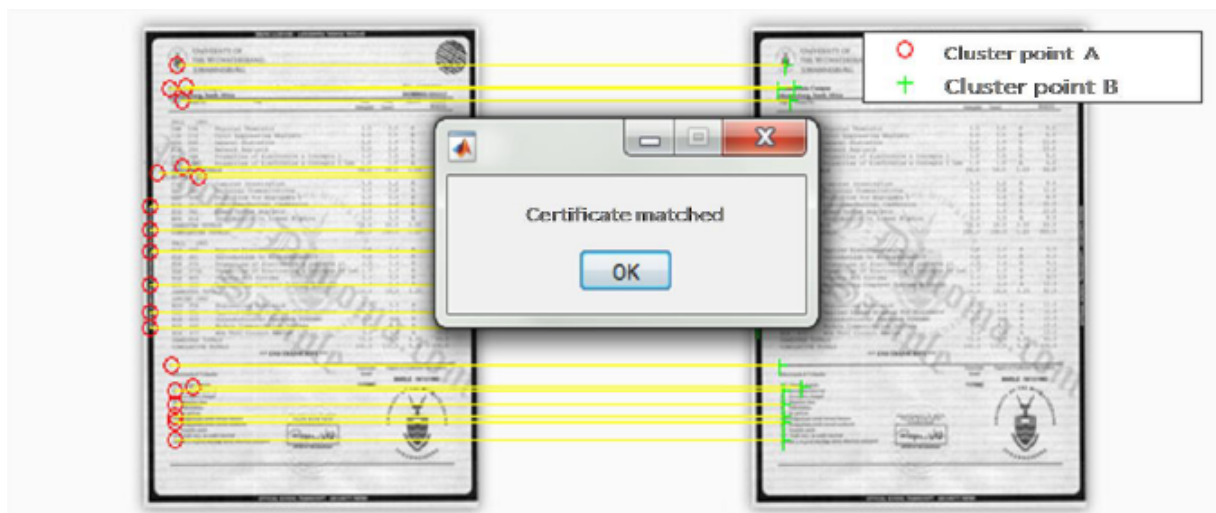
O *dataset* utilizado pelos autores consiste em mais de vinte e quatro mil amostras de documentos não rotulados, oficialmente emitidos por duas universidades: Enugu State University of Science and Technology e University of California Irvine.

A metodologia dos autores utiliza o algoritmo K-means como técnica principal para descobrir os padrões dominantes em certificados acadêmicos. Dessa forma, o processo de treinamento consiste na aplicação desse algoritmo sobre as características extraídas dos documentos do *dataset*, com o objetivo de agrupá-los em dezesseis grupos. O algoritmo inicializa centroides e atribui iterativamente cada arquivo ao centroide mais próximo usando um modelo de equidistância. Os centroides são atualizados através do cálculo das médias dos pontos pertencentes a cada *cluster* até atingir convergência. Para a verificação

de documentos, o sistema extrai suas características e calcula distâncias em relação aos centroides estabelecidos, classificando-o como legítimo, quando próximo de algum padrão conhecido, ou suspeito, quando distante de todos os *clusters*.

Embora os autores não especifiquem os processos de extração de características, as figuras apresentadas sugerem fortemente o uso de *features* visuais. As imagens, como exemplificado na Figura 19, mostram marcações circulares em pontos específicos dos certificados, destacando elementos como logos institucionais e aspectos estruturais dos documentos. Assume-se, portanto, que o sistema captura e converte essas informações em vetores numéricos compatíveis com o K-means.

Figura 19 – Resultado da Verificação de um Documento Autêntico



Fonte: (Mohammed; Nwobodo; Ekene, 2024)

O modelo foi validado experimentalmente com certificados reais e demonstrou capacidade de distinguir documentos autênticos de falsificados, já que os resultados reportaram uma acurácia geral de 86,53%. Os autores destacam como principal vantagem a capacidade do sistema de operar efetivamente com *datasets* limitados, característica importante em cenários reais onde a disponibilidade de dados rotulados é restrita por questões éticas e de privacidade.

4 TRABALHOS BASE

Este capítulo apresenta os trabalhos que constituem a base técnica da solução desenvolvida.

4.1 APRENDIZADO DE CARACTERÍSTICAS VISUAIS SEM SUPERVISÃO

Para identificar famílias de documentos, faz-se necessária a extração de características visuais de cada página. Esse papel é desempenhado pelo modelo DINOv2, proposto por Oquab et al. (2024). O trabalho busca desenvolver uma arquitetura capaz de extrair características visuais de propósito geral, cujos autores explicam como representações que descrevem qualquer imagem de forma suficientemente rica para serem aproveitadas diretamente em tarefas diversas, como classificação, segmentação e estimativa de profundidade, sem qualquer adaptação para o objetivo de destino. Para alcançá-lo, o método dispensa rótulos e aprende de forma auto-supervisionada a partir unicamente da estrutura das próprias imagens.

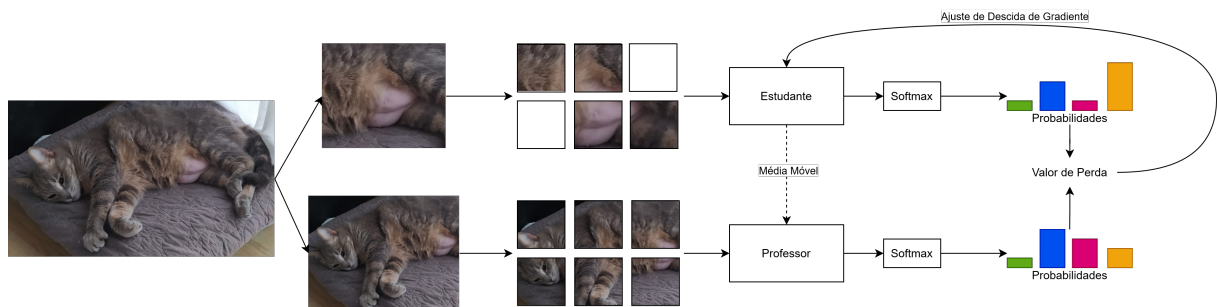
Em termos de arquitetura, o DINOv2 é um *vision transformer* que, conforme exposto na fundamentação teórica, divide a imagem em fragmentos regulares de tamanho fixo, os *patches*, e os processa como uma sequência de *tokens*, cujas relações são estabelecidas pelo mecanismo de atenção. O modelo treinado, com cerca de 1,1 bilhão de parâmetros, é em seguida comprimido em versões menores que preservam suas representações sem repetir o treinamento completo. Um desses artefatos é utilizado neste TCC, conforme detalhado no capítulo seguinte.

Esse aprendizado sem rótulos concretiza-se por uma técnica de autodestilação, que parte da premissa de que porções diferentes de uma mesma imagem mostram o mesmo conteúdo e, por isso, devem levar à representações similares. O método consiste em manter duas cópias do modelo, chamadas estudante e professor, e extraem-se de cada imagem diversos recortes, em que o professor processa vistas que correspondem às cenas inteiras, e o estudante processa vistas de detalhes locais. Cada um converte o recorte que recebe em uma distribuição de probabilidades sobre um conjunto fixo de protótipos, isto é, padrões de referência aprendidos ao longo do próprio treinamento, que indica em que medida cada padrão está presente no recorte. Mede-se então o quanto a distribuição que o estudante obtém do detalhe se afasta da que o professor obtém da cena inteira, e os pesos do estudante são ajustados, por descida de gradiente, para reduzir esse afastamento a cada iteração. Em outras palavras, o estudante aprende a reconhecer, a partir de um único detalhe, o mesmo conteúdo que identificaria na imagem toda. O professor, por sua vez, não é treinado diretamente, e seus pesos são apenas uma média móvel daqueles que o estudante assume ao longo do treinamento, isto é, desloca os pesos um pouco na direção dos do estudante, acumulando ao longo do treinamento uma versão suavizada dele. Por mudar lentamente, oferece ao estudante um alvo estável, em vez de uma referência que

saltaria a cada iteração.

Soma-se ao processo de treinamento uma comparação feita em nível de *patch*, em que parte dos *patches* é ocultada da entrada do estudante, enquanto o professor continua recebendo a imagem completa. Para cada região escondida, o professor produz uma representação a partir da imagem inteira, e o estudante deve reproduzi-la usando apenas os *patches* visíveis ao redor, em um exercício de preenchimento de lacunas. O ajuste segue o mesmo princípio de antes, medindo o quanto as duas representações se afastam e corrigindo os pesos do estudante por descida de gradiente. Assim, enquanto a primeira comparação molda a representação global da imagem, esta segunda torna cada *token* de *patch*, por si só, descritivo de sua região. A figura a seguir diagrama, de maneira simplificada, todo o processo.

Figura 20 – Diagrama de Treinamento DINO



Fonte: o autor

Ao final, cada imagem processada dá origem a dois tipos de representação, um vetor global, associado a um *token* de classe que resume a imagem como um todo, e a grade de *tokens* de *patch*, em que cada vetor descreve sua região no contexto das demais. O presente trabalho de TCC aproveita essa capacidade de descrição regional para alimentar a descoberta de *templates*.

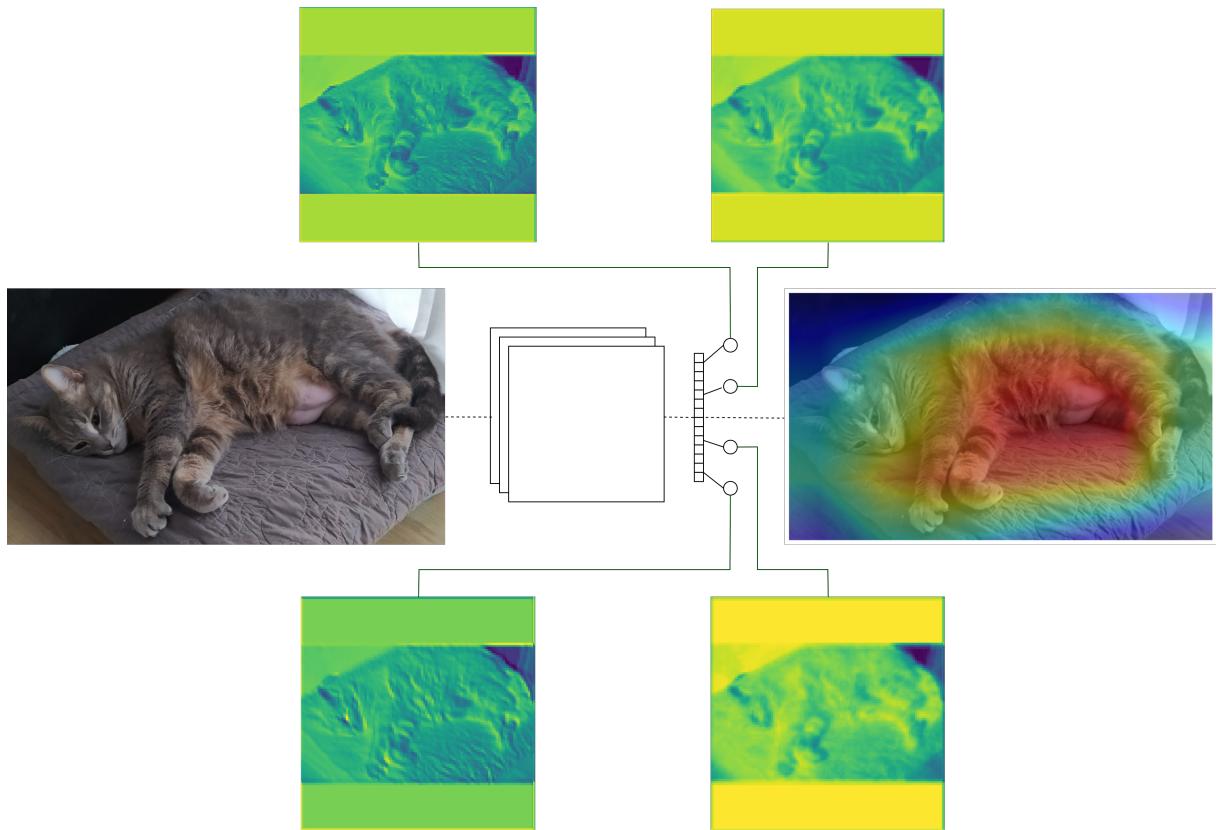
4.2 AGREGAÇÃO REGIONAL DE CARACTERÍSTICAS VISUAIS

O trabalho de Tolias; Sivic; Jégou (2016) explora o problema da recuperação de objetos específicos, temática da área de visão computacional que consiste em localizar um mesmo objeto entre diversas imagens, ainda que este apareça sob diferentes escalas, posições ou ângulos. Os autores propõem um método para descrever uma foto a partir de sua redução a uma representação numérica comparável. Esse descritor é obtido do resultado dos mapas de ativação de uma rede neural convolucional, e agrega informações de várias regiões da imagem em um único vetor compacto, assim, o problema da busca por um objeto é simplificado a medir a similaridade entre *embeddings*. Essa abordagem é eficiente porque a extração de todas as características acontece em uma só passagem pela

rede, sem que seja necessário o recorte e processamento de sub-imagens, como abordagens anteriores faziam.

O ponto de partida do método é a operação de ativação máxima de convoluções (MAC), técnica utilizada para visualizar as características aprendidas por uma rede convolucional, o descritor em si. Conforme exposto na fundamentação teórica, uma CNN processa uma imagem através de camadas convolucionais, que atuam como filtros aprendíveis de características – detectores de padrões visuais específicos, como texturas, formas ou partes de objetos. Descartada a cabeça classificadora da rede, o resultado de um processamento é dado a partir da saída da última camada convolucional e passa a ser um tensor de dimensões $W \times H \times K$, onde K corresponde ao número de mapas de características e (W, H) às dimensões de cada mapa de características, isto é, cada um dos K canais corresponde a um filtro previamente aprendido e constitui, em si, um mapa bidimensional que indica em quais posições da imagem aquele padrão foi encontrado. MAC toma, para cada canal, sua ativação máxima ao longo de todas as posições espaciais, produzindo um vetor de K dimensões em que cada componente registra a intensidade com que o padrão correspondente aparece na imagem. Por tomar o máximo sobre todas as posições, a representação é invariante a translações, mas descarta a informação de onde cada padrão ocorre. A figura seguinte ilustra esse processo, com o processamento de uma imagem por uma CNN (a mesma já exemplificada durante o Capítulo 2), tomam-se alguns exemplos de saídas de canais e a visualização da operação MAC resultante, como mapa de calor.

Figura 21 – Representação de MAC



Fonte: o autor

Para reintroduzir no descritor as informações de localização previamente descartadas, os autores propõem uma extensão regional do MAC, que denominam *Regional MAC* (R-MAC). Para isso, sobre o mapa de ativações amostram-se regiões quadradas em múltiplas escalas, sobrepostas entre si e de tamanho decrescente. A cada região aplica-se o *max-pooling* para obter um vetor regional, então, sobre cada vetor, são aplicadas operações de normalização e de branqueamento via Análise de Componentes Principais (PCA-whitening) – algoritmo que descorrelaciona os canais e equilibra suas contribuições, evitando que valores de canais mais expressivos dominem os de canais menos expressivos. Por fim, os vetores regionais são somados em um único, normalizado ao final, cuja dimensão permanece igual a K , independentemente de quantas regiões foram amostradas. Em termos mais amplos, cada vetor regional descreve o conteúdo de uma porção específica da imagem, de modo que o conjunto de regiões registra tanto a intensidade da ocorrência de padrões quanto a parte da imagem em que aparecem, então, ainda que a soma final reúna tudo em um único descritor, a caracterização independente de cada região o mantém sensível à disposição espacial das características aprendidas, sem abrir mão da tolerância a pequenas translações que o *max-pooling* de cada região preserva.

Figura 22 – Representação de R-MAC



Fonte: o autor

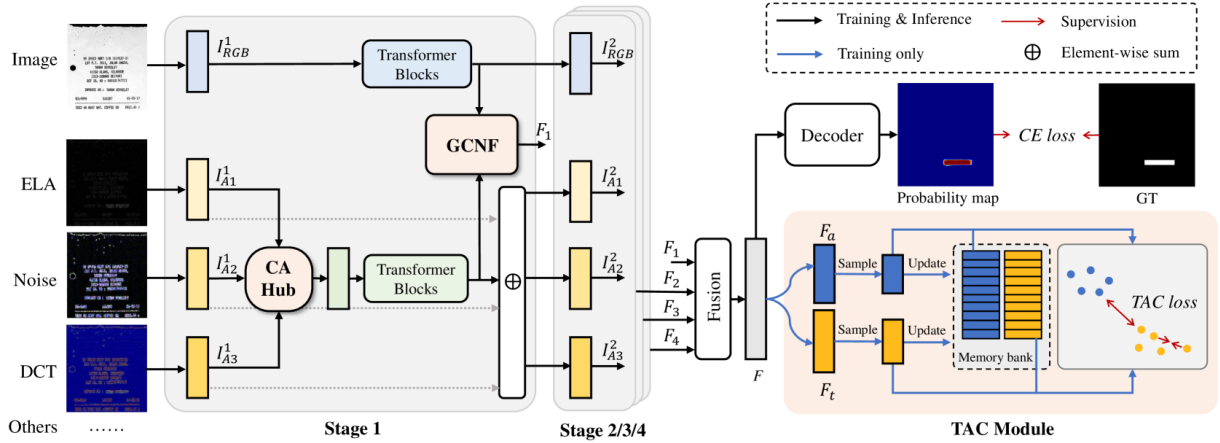
Assim, de maneira geral, o R-MAC constrói descritores de imagens a partir de detectores de padrões visuais aprendidos e da região em que esses padrões aparecem. Essa construção torna a comparação desses descritores uma busca por objetos, e não apenas por imagens parecidas: fotos com um mesmo objeto, disposto na mesma região, acionam o mesmo conjunto de detectores e resultam em *embeddings* matematicamente próximos, enquanto objetos distintos e/ou em diferentes regiões geram vetores gradualmente mais distantes. A Figura 22 ilustra esse efeito com duas fotografias distintas de um mesmo gato, em fundos e poses diferentes. O retângulo azul destaca uma região equivalente nas duas imagens (a barriga do gato), que recai sobre o mesmo objeto (o gato), e é dessa correspondência que resultam descritores de alta similaridade, apesar das diferenças de cenário, iluminação e enquadramento. É essa propriedade que o presente trabalho aproveita, não para reencontrar um objeto, mas para reconhecer a estrutura de um documento, isto é, o descritor de cada documento passa a funcionar como uma impressão digital de seu *layout*, de modo que documentos construídos sobre um mesmo modelo, com a mesma disposição de cabeçalho, blocos de texto e selos, produzem descritores próximos e podem ser agrupados. Por fim, vale observar que, embora o R-MAC tenha sido originalmente definido sobre ativações convolucionais, o mesmo princípio de agregação se aplica a qualquer grade de características, como os *tokens* de um *vision transformer*, opção adotada neste trabalho e detalhada posteriormente na metodologia.

4.3 DETECÇÃO DE ADULTERAÇÃO DE TEXTO EM DOCUMENTOS

O trabalho de Luo et al. (2024) propõe um detector voltado à identificação e localização de adulterações textuais em imagens de documentos, denominado *Asymmetric Stream Contrastive Transformer* (ASC-Former). Ao invés de classificar o documento como um todo, o método trata o desafio como um problema de segmentação: produz um mapa em nível de *pixel* que infere quais regiões da imagem foram manipuladas. Além desse

framework, os autores disponibilizam o conjunto de dados *Real Text Manipulation* (RTM), que possui mais de quatorze mil imagens de documentos autênticos e adulterados tanto manualmente quanto por processos automáticos.

Figura 23 – *Pipeline ASC-Former*



Fonte: (Luo et al., 2024)

A premissa central do método é de que um trecho de texto adulterado, por mais bem-executado que seja, rompe a consistência que deveria manter com o texto autêntico ao seu redor no que diz respeito a propriedades como tipografia, espaçamento, ruído, marcas de compressão ou demais vestígios estatísticos. Para capturar esses desvios, o ASC-Former, cujo *pipeline* completo é ilustrado na Figura 23, processa o documento em dois fluxos assimétricos, isto é, partem de entradas de naturezas distintas. O primeiro recebe a imagem original e em cores (na figura, fluxo que parte de Image), enquanto o segundo recebe uma representação forense composta por três transformações aplicadas à imagem (na figura, fluxo que parte de ELA, Noise e DCT):

- utiliza-se *Error Level Analysis* (ilustrado por ELA, no diagrama) para recomprimir a imagem a uma taxa de erro conhecida e medir, região a região, o quanto ela se altera nesse processo, já que regiões autênticas e adulteradas reagem com sensibilidades distintas a essa segunda compressão e podem denunciar uma edição;
- *Steganalysis Rich Model* é utilizado para extrair o ruído da imagem (ilustrado por Noise, no diagrama), isto é, da textura de alta frequência introduzida pela captura da imagem, seja de uma câmera ou *scanner*. Como esse ruído funciona como uma assinatura própria da captura, uma região trazida de outra origem ou sintetizada carrega um padrão incompatível com o do restante do documento;
- aplica-se a transformada discreta de cosseno por blocos para representar a imagem no domínio da frequência (ilustrado por DCT, no diagrama), onde podem ser evidenciados dois indícios: a adulteração tende a romper o padrão de blocos deixado pela

compressão da imagem e, ao mesmo tempo, o traçado do texto aparece realçado, o que favorece a análise sobre o próprio conteúdo textual.

A combinação de todas as informações ocorre em dois momentos sucessivos, iniciando pelo fluxo forense. Suas três transformações oferecem visões complementares da mesma imagem, e o módulo de agregação sensível à consistência (na figura, CA Hub) examina cada uma isoladamente, medindo o quanto a resposta de uma mesma transformação varia de uma região para outra do documento. A premissa é de que uma transformação cuja resposta varia fortemente entre regiões está reagindo a vestígios de adulterações. A partir dessas variações, resumidas região a região por estatísticas como média, variância, máximo e mínimo, o módulo aprende a ponderar as três transformações e a normalizar seus pesos, de modo que seu resultado é uma única representação que privilegia a região mais informativa.

Consolidado o sinal forense, ele e a imagem visível seguem, cada um, por um codificador *transformer*. Como visto na fundamentação teórica, esse tipo de rede divide a entrada em pequenas regiões e usa o mecanismo de atenção para relacioná-las, descrevendo cada uma no contexto das demais. Cada codificador produz, assim, mapas de características em diferentes escalas e dimensões, dos detalhes finos de um caractere ao contexto de um bloco inteiro. Em cada escala, um módulo de fusão (na figura, GCNF) combina os mapas visível e forense por meio de um mecanismo de atenção de vizinhança, que confronta cada região apenas com a janela ao seu redor sob um parâmetro aprendido, que dosa quanto a evidência forense pesa na fusão. Isso se diferencia do mecanismo de atenção do codificador, que é capaz de alcançar regiões por toda a imagem.

Os mapas fundidos nas diferentes escalas são então reunidos em uma única representação (na figura, passam pelo bloco Fusion e resulta em F), o que permite tratar tanto a adulteração de um único caractere quanto a substituição de um campo inteiro. É sobre F que se apoiam tanto o treinamento do modelo quanto, depois, a geração do mapa final.

Durante o treinamento, F alimenta um módulo de aprendizado contrastivo (na figura, TAC Module), cujo objetivo é tornar mais discriminativas as características de regiões adulteradas e autênticas: ele aproxima as representações de mesma natureza e afasta as de naturezas opostas, acentuando a fronteira entre o genuíno e o forjado. Como em um documento os trechos adulterados são muito mais raros que os autênticos, o módulo equilibra a amostragem das duas classes e mantém um banco de memória com exemplos de lotes anteriores, garantindo amostras suficientes da classe mais rara.

Finalmente, durante a inferência, um decodificador (na figura, Decoder) lê a representação F e produz o mapa de adulteração em nível de *pixel* que estima, para cada região do documento, a probabilidade de ter sido manipulada. É esse o sinal de detecção primário da solução proposta: ajustado sobre as falsificações sintéticas geradas neste trabalho, o ASC-Former aprende a evidenciar as costuras sutis deixadas pela edição de campos em um documento legítimo, enquanto a agregação desse mapa em um escore por documento

fecha a decisão.

5 ARQUITETURA DA SOLUÇÃO

Este capítulo descreve o processo de concepção da solução proposta, percorrendo cada módulo da arquitetura final e expondo o racional das escolhas de projeto que a conformaram.

5.1 DA ABORDAGEM INICIAL À SOLUÇÃO ATUAL

A concepção inicial deste TCC partia de uma abordagem inteiramente não-supervisionada. A ideia era extrair de cada documento três representações complementares de características – visual, textual e estrutural –, combiná-las em uma representação única, projetá-la em um espaço vetorial e calcular agrupamentos, que representariam padrões de cada tipo de documento. Assim, sob a premissa de que um documento falsificado destoaria em um espaço vetorial multidimensional, detectores de anomalia mediriam o quanto cada nova submissão se desviaria do padrão estabelecido por cada grupo.

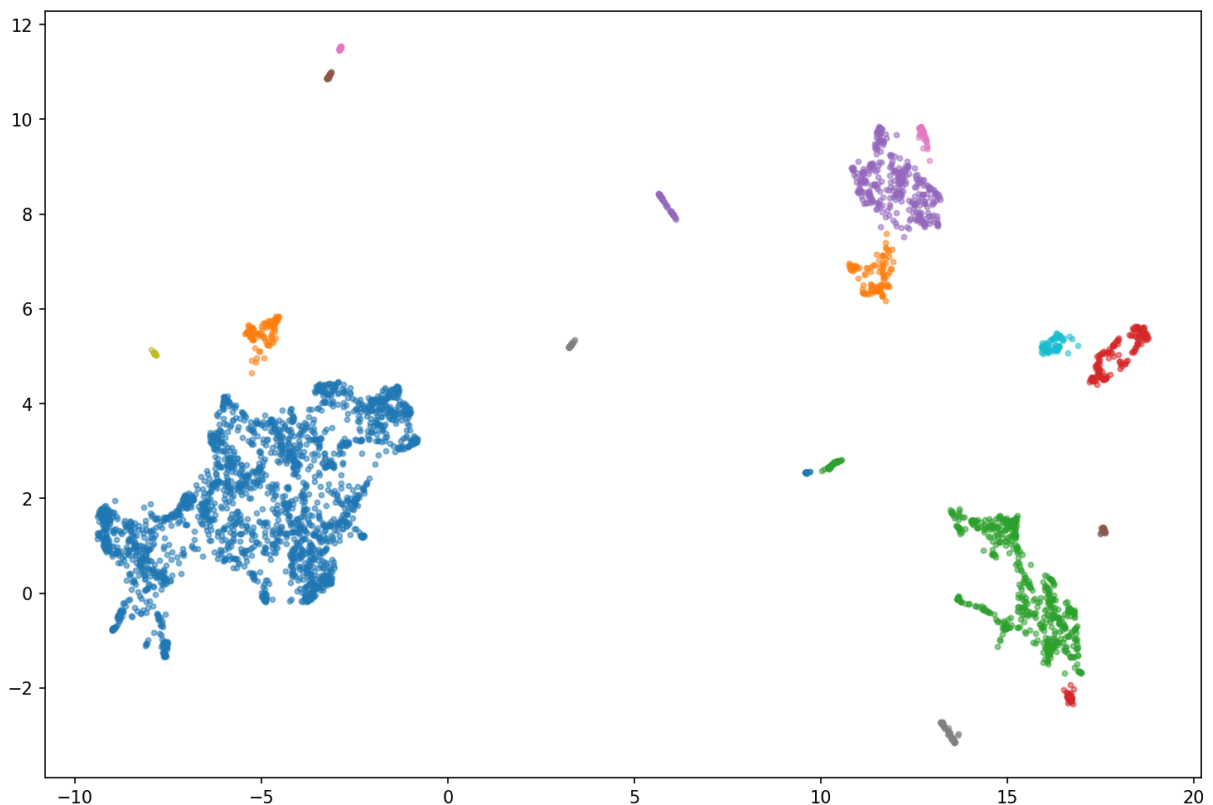
A primeira dessas representações a se mostrar inviável foi a textual. A tentativa inicial extraía o texto dos documentos por OCR e empregava um *large language model* (nesse caso, foi utilizado o Qwen 2.5 com 7 bilhões de parâmetros) para identificar e padronizar o nome da instituição emissora, consolidando suas variantes em um pseudo-rótulo, na expectativa de que houvessem padrões distinguíveis dos documentos de cada instituição que auxiliassem no processo de agrupamento. O resultado, contudo, mostrou-se muito ruidoso, e o esforço levou a reexaminar a própria premissa do módulo, assentada na falsa hipótese de que uma fraude textual introduziria anomalias estatísticas mensuráveis, como desvios na distribuição de termos ou na formatação de números. No entanto, como a falsificação reaproveita um documento legítimo e nele substitui apenas campos pontuais, aquilo que efetivamente muda no texto – nomes, datas, notas e códigos – é justamente o que já varia de um documento autêntico para outro, assim, qualquer processo de extração trataria isso como ruído, e não como sinal. Sem desvio estatístico que denunciasses a fraude, esse módulo foi abandonado em definitivo.

As representações visual e estrutural, reunidas no esforço de descrever a aparência e a organização de cada documento, respectivamente, revelaram-se o maior obstáculo. Ainda que disponha de documentos de naturezas variadas, estima-se que o *dataset* obtido não ultrapasse dezenas de *templates* distintos, cada um com vários exemplares – de modo que o desafio nunca foi a falta de exemplos, mas a grande variação dentro de uma mesma família: parte dos documentos é digitalizada por *scanner* e parte é fotografada, e mesmo modelos idênticos exibem pequenas diferenças de disposição, como selos e assinaturas posicionados de forma levemente distinta. Dessa forma, o problema central tornou-se obter uma representação que mantivesse próximos, no espaço vetorial, documentos de uma mesma família apesar dessas variações.

Diversos extratores pré-treinados foram avaliados para a tarefa. Redes convolucio-

nais treinadas sobre imagens naturais, como a EfficientNet-B0 mostraram-se inadequadas, pois seu pré-processamento recorta a região central da imagem e descarta as bordas, regiões geralmente distintivas, dado que possuem cabeçalhos, logotipos e selos. Ainda, as características que essas redes aprendem a reconhecer durante o pré-treinamento, voltadas aos objetos e texturas de fotografias do mundo real, pouco revelam sobre as de um documento. Assim, a aplicação de DBSCAN sobre as extrações não produziu *clusters* coerentes e uniformes, pelo contrário, produziu agrupamentos que concentravam diversos tipos de documentos. Mesmo os *clusters* mais concentrados sofreram do mesmo problema, prova de que o modelo não foi capaz de extrair características relevantes. O espaço vetorial resultante da melhor extração pode ser visto na Figura 24, que ilustra também os diferentes grupos formados, em diferentes cores.

Figura 24 – DBSCAN Sobre Extrações EfficientNet-B0

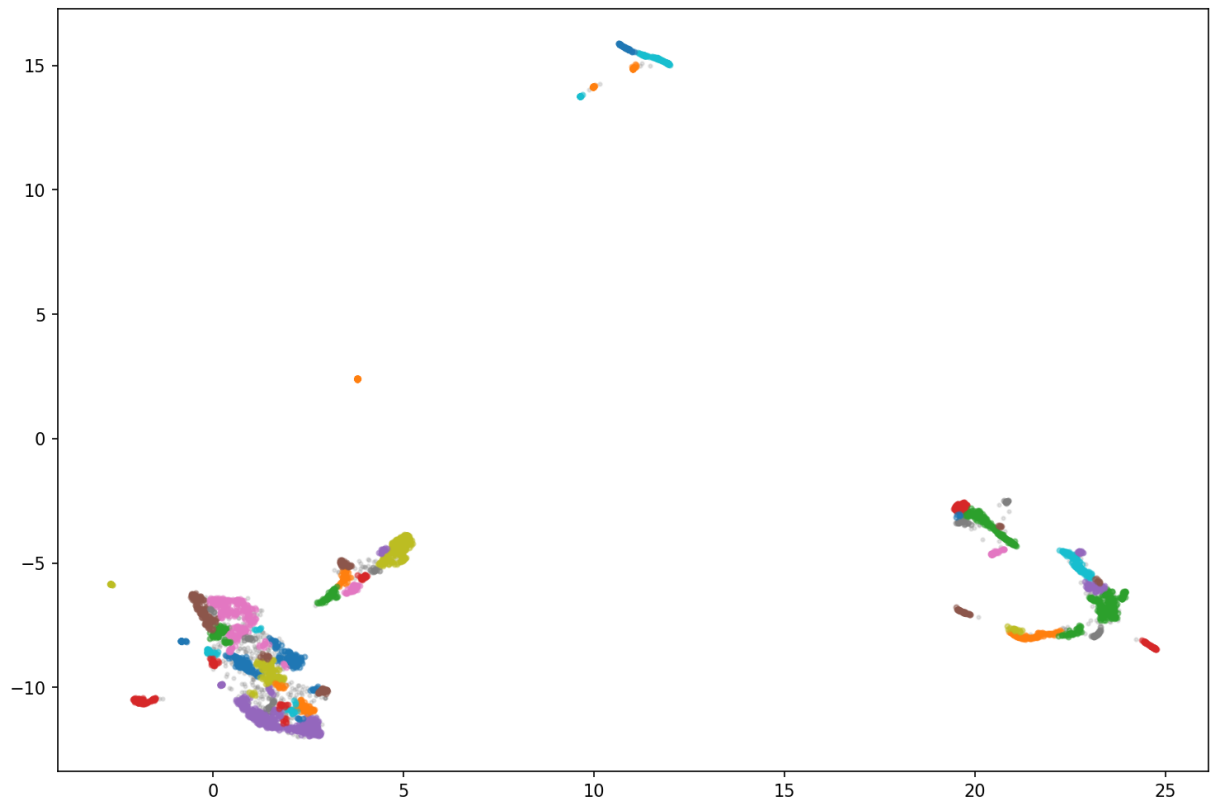


Fonte: o autor

A abordagem evoluiu para o uso de *transformers*. Primeiro, tentou-se a utilização do *Document Image Transformer* (DiT), modelo pré-treinado especificamente sobre documentos, que tampouco resolveu o problema. A falha se deve não à arquitetura, mas ao método e ao objetivo do treinamento do modelo. O DiT aprende por reconstrução de trechos ocultos da imagem, tarefa que o leva a aprender aquilo que os documentos têm em comum (texto escuro sobre fundo claro, margens regulares etc.), e não aquilo

que os distingue. O resultado foi um espaço em que os documentos se comprimiram em faixas estreitas de similaridade, como ilustra a Figura 25, apresentando o mesmo problema previamente encontrado.

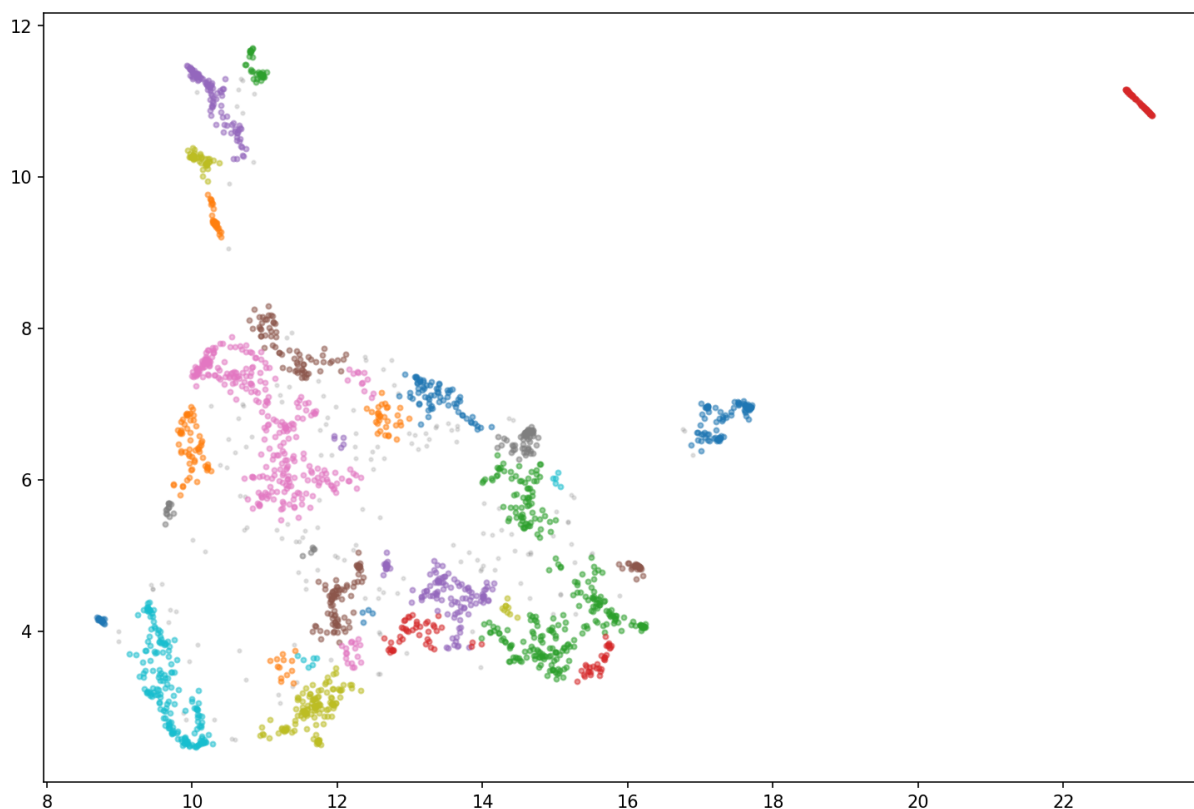
Figura 25 – DBSCAN Sobre Extrações DiT



Fonte: o autor

O melhor resultado surgiu com o uso do modelo DINOv2, um *vision transformer* de mesma natureza arquitetural, porém pré-treinado de forma auto-supervisionada sob um objetivo discriminativo, que explicitamente aproxima as representações de imagens semelhantes e afasta as de imagens distintas. O contraste entre os dois deixou evidente que, para essa tarefa, o objetivo de treinamento pesa mais do que o próprio domínio dos dados, em outras palavras, o DiT (treinado sobre documentos) perdeu para o DINOv2 (treinado sobre imagens genéricas) porque o que faltava não era familiaridade com documentos, mas capacidade e aprendizado para discriminá-los.

Figura 26 – DBSCAN Sobre Extrações DINOv2



Fonte: o autor

Ainda assim, a agregação ingênua dos *tokens* do DINOv2 (pela média global da imagem ou por uma divisão em poucos quadrantes) também não produziu um espaço vetorial coerente, embora visivelmente mais harmonioso, como ilustra a Figura 26. Nesse caso, as variações de captura e de disposição dispersavam documentos de uma mesma família que deveriam permanecer próximos. A saída foi recorrer à agregação regional de características (R-MAC) apresentada no capítulo anterior, cujo *max-pooling* por região confere tolerância às pequenas translações de selos e assinaturas. O espaço resultante permaneceu esperso, mas, para o objetivo de agrupar documentos por *template*, passou a funcionar de forma satisfatória, e é dessa extração, dita de *layout*, que a solução final se vale.

Além das dificuldades de extração, um obstáculo independente comprometia o cerne da abordagem original, isto é, a detecção de anomalia pressupõe algum referencial daquilo que se considera atípico para calibrar seus limiares, e dentro do *dataset*, supõe-se que todos os documentos são válidos. Sem exemplos de fraude, era impossível ajustar o ponto de corte de cada detector ou aferir, ainda que de forma aproximada, a qualidade da classificação.

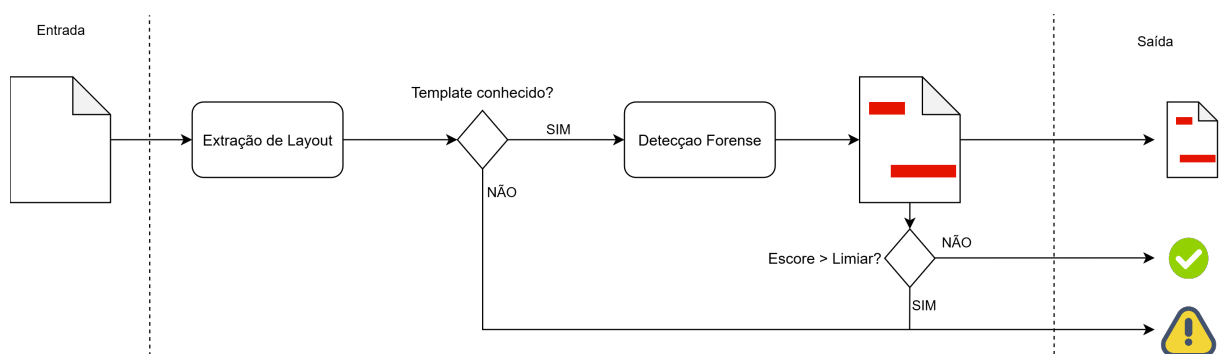
Diante desse quadro, a solução foi redirecionada e a extração de *layout* seguida de clusterização, longe de ser abandonada, foi reaproveitada como ponte para a abordagem

definitiva. Em vez de servir diretamente à classificação, passou a cumprir três papéis de suporte: descobrir as famílias de *template* presentes no *dataset*; semear a geração de falsificações cientes do *template* de cada documento, que suprem a ausência de exemplos rotulados, fornecendo o material de treino supervisionado que faltava; e prover, em tempo de inferência, um *gate* que sinaliza como suspeito todo documento não atribuível a nenhuma família conhecida. O núcleo da detecção, por sua vez, deslocou-se para o detector forense supervisionado apresentado no capítulo anterior (ASC-Former), treinado sobre essas falsificações para localizar, em nível de *pixel*, os vestígios deixados pela edição.

5.2 VISÃO GERAL DA ARQUITETURA

Conforme apresentado na Figura 27, a solução desenvolvida encadeia dois sinais que partem de um mesmo documento e atuam em sequência. O primeiro deriva da extração de *layout*, em que o descritor R-MAC de cada documento é confrontado com as famílias de *template* descobertas pelo agrupamento e, quando não se associa a nenhuma delas, dispara um *gate* de pertencimento. O segundo é produzido por um ASC-Former retreinado sobre as falsificações geradas neste trabalho, que percorre o documento em nível de *pixel* e devolve um mapa de calor estimando, para cada região, a probabilidade de manipulação. Esse mapa é então condensado a um *score* e é comparado a um limiar calibrado. Essa arquitetura corresponde ao modelo de ameaça assumido, no qual um fraudador não constrói um documento do zero, mas reaproveita um exemplar legítimo como base e nele substitui campos pontuais.

Figura 27 – Representação do Fluxo de Inferência



Fonte: o autor

Por um lado, como explicado no capítulo anterior, substituir um campo em um documento já digitalizado exige reescrevê-lo e reinseri-lo sobre a imagem, e isso deixa uma pista: o trecho editado rompe a continuidade que mantinha com o entorno autêntico no traçado da tipografia, no ruído de captura e nas marcas deixadas pela compressão, vestígios que as transformações do detector forense são capazes de realçar. Por outro lado,

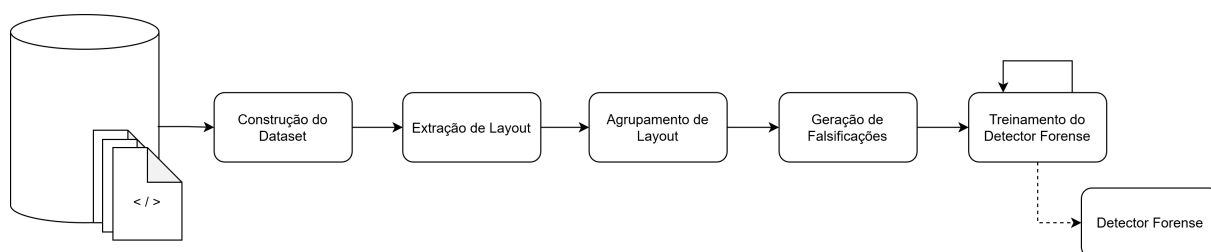
caso a falsificação não se apoie em nenhum *template* conhecido, não se pode mais garantir que o mesmo modelo de ameaça seja válido, fazendo-se necessária a revisão de um humano especialista. É então o *gate* de *layout* que a captura, ao constatar que sua estrutura não corresponde a nenhuma família aprendida. As duas avaliações cobrem, assim, superfícies de ataque complementares, onde o detector forense flagra a edição feita sobre um *template* aprendido, enquanto o *gate* rejeita documentos desconhecidos, legítimos ou não.

Convém, antes de detalhar os fluxos de treinamento e inferência, fixar a terminologia que distingue as duas capturas, evitando uma ambiguidade recorrente. Chama-se aqui de *layout* a impressão digital estrutural do documento – o descritor que resume a disposição de cabeçalhos, blocos de texto e selos, sobre o qual a extração R-MAC e agrupamento operam. Reserva-se o termo visual, ou forense, à visão em nível de *pixel* que examina como o documento foi renderizado, expondo as inconsistências locais de edição que alimentam o detector ASC-Former.

5.3 TREINAMENTO

Esta seção trata de explicar detalhadamente o fluxo de treinamento, representado pela Figura 28: os documentos extraídos e convertidos em imagem alimentam a extração de *layout* e o agrupamento, cujos *clusters* orientam a geração das falsificações que, por fim, retreinam o detector forense. Embora boa parte dos módulos seja compartilhada entre treinamento e inferência, convém, por clareza, detalhá-los ao longo desta seção, de modo que a seção de inferência possa se ater ao encadeamento dos sinais, sem reapresentar cada módulo.

Figura 28 – Representação do Fluxo de Treino



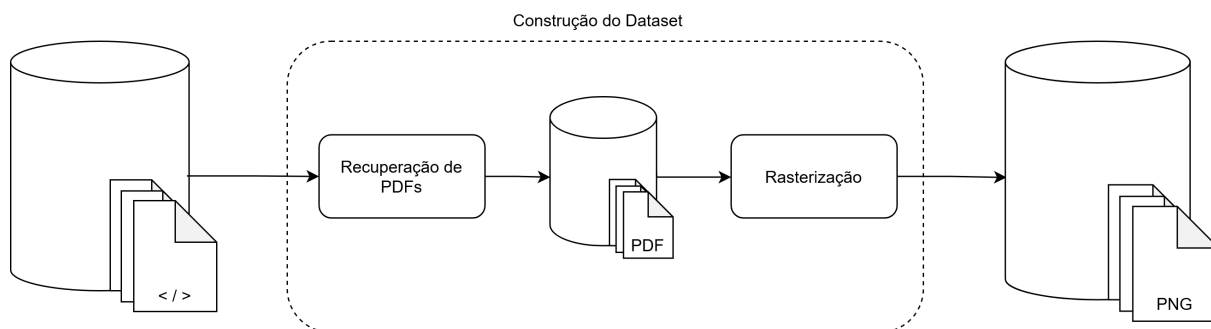
Fonte: o autor

5.3.1 Construção do Dataset

O *dataset* que sustenta este trabalho foi construído a partir de 2161 arquivos fornecidos pela UFSC, todos no formato XML do esquema de diploma digital do Ministério da Educação. Desses, 2066 traziam ao menos um documento embutido, dos quais se selecionou o material de interesse deste trabalho, que são os PDFs da documentação comprobatória de etapas ou cursos anteriores, isto é, históricos escolares e certificados

de conclusão de ensino médio, técnico ou superior. Foi desse acervo que se obteve, em três passos sucessivos, o conjunto de imagens efetivamente utilizado nas demais fases: a recuperação dos PDFs, sua conversão em imagens e a deduplicação do resultado.

Figura 29 – Detalhamento do Módulo de Construção de Dataset

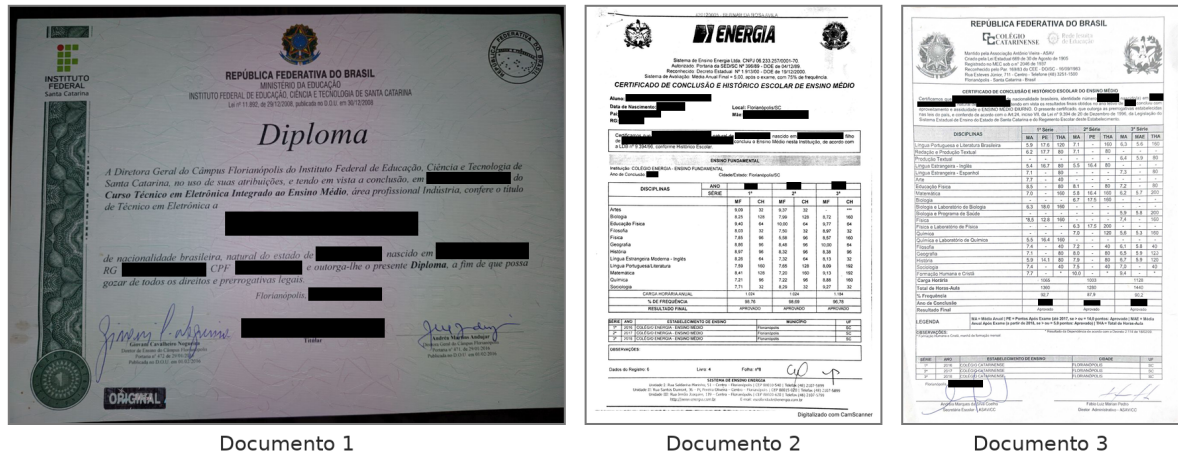


Fonte: o autor

A recuperação dos PDFs começou pela identificação, dentro de cada XML, dos elementos que carregavam os documentos de interesse. No entanto, seu conteúdo não se deu em referência a um arquivo ou em formato binário direto, mas unicamente ou duplamente codificado em base64. Isso exigiu detectar a profundidade da codificação antes de decodificar os arquivos, visto que a primeira decodificação ora revelava diretamente os *bytes* do PDF, ora produzia um novo texto em base64, que então era decodificado de novo. Ao fim, cada documento de interesse passou a existir como um arquivo PDF independente.

Cada PDF recuperado foi então rasterizado em imagens, uma por página, a 200 *dots per inch*, em cores e no formato PNG sem perdas, o que produz aproximadamente 1650×2330 *pixels* por página A4. Essa resolução preserva os detalhes finos, enquanto o formato sem perdas evita os artefatos de compressão que poderiam, mais tarde, ser confundidos com vestígios de manipulação. Optou-se por converter todas as páginas, e não apenas a primeira, pois parte dos documentos traz informações relevantes além da folha de rosto (como as páginas internas de um histórico ou o verso de um diploma), de modo que cada página passou a ser um item independente do *dataset*. A Figura 30 reúne três exemplos de imagens que conformaram o *dataset* final, com seus dados sensíveis anonimizados.

Figura 30 – Amostras do Dataset



Fonte: o autor

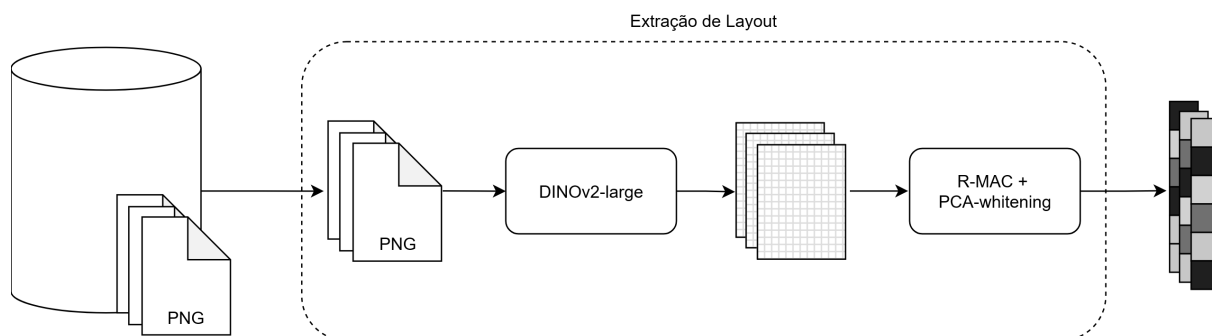
Por fim, o conjunto formado passou por uma deduplicação. Como um mesmo PDF podia ter sido inserido mais de uma vez em um envelope XML ou reenviado em arquivos distintos, parte das imagens era idêntica a outra já presente. Tais repetições foram identificadas e descartadas, mantendo-se um único representante de cada grupo. Das 2105 imagens inicialmente geradas, 116 foram removidas nesse passo, e as 1989 restantes constituem o *dataset* final sobre o qual operam todas as fases seguintes do trabalho.

Vale ressaltar que esse conjunto final é internamente heterogêneo. Seus documentos provêm de instituições de ensino diversas e se distribuem por muitos *templates* distintos, aos quais se somam ainda variações de um mesmo modelo, por vezes decorrentes de revisões ao longo do tempo. Ao todo, não se sabe quantas famílias de *template* existem, mas com a extração e agrupamento de *layout* descritos a seguir, estima-se que há algumas dezenas delas, com um número de exemplares que varia drasticamente de uma para outra, já que certos modelos reúnem dezenas ou centenas de documentos, enquanto outros possuem apenas algumas unidades.

5.3.2 Extração de Layout

Construído o *dataset* de imagens, o primeiro processamento extrai um descritor de *layout* de cada documento, ou seja, um vetor que resume sua estrutura visual e o torna comparável aos demais. Esse extrator combina dois componentes já apresentados, o DINOv2 como codificador, seguido da agregação regional por R-MAC.

Figura 31 – Detalhamento do Módulo de Extração de Layout



Fonte: o autor

Recapitulando, *vision transformers* não processam uma imagem *pixel a pixel*, eles a repartem em uma grade de pequenos blocos quadrados, os *patches*, e convertem cada um em um vetor, chamado *token*. Neste caso, para que cada imagem se divida em um número inteiro de *patches*, elas são redimensionadas e preenchidas até 1036×1036 *pixels*, ou seja, 74 blocos de 14 *pixels* de lado. O módulo utiliza o modelo DINOv2-large congelado – isto é, aproveitado exatamente como foi pré-treinado e disponibilizado – para produzir uma grade de 74×74 *tokens*, assim, cada um dos vetores de 1024 dimensões resultantes descreve a região de 14×14 *pixels* correspondente. Essa granularidade proporciona precisão espacial suficiente para capturar elementos importantes, como logotipos e selos.

Sobre a grade de *tokens* produzidos, o módulo aplica o R-MAC conforme método do artigo original de Tolias; Sircé; Jégou (2016), em três escalas, isto é, com janelas quadradas de três tamanhos, em que a maior cobre a grade inteira, e as menores, sobrepostas entre si, recortam porções cada vez mais locais da página, totalizando em torno de 22 regiões. Assim, o descritor capta desde a organização geral do documento até seus elementos pontuais. Então, é aplicado o *max-pooling* sobre cada uma das regiões: para cada uma das 1024 dimensões, toma-se o maior valor entre os *tokens* contidos na região, de modo que um logotipo ou selo que ative fortemente uma ou mais posições sobreviva à agregação em vez de ser diluído. Em seguida, os vetores regionais resultantes, normalizados, são somados elemento a elemento em um único vetor final de 1024 dimensões por documento.

Por fim, é aplicado o PCA-whitening que, ao contrário das anteriores, é uma transformação que precisa ser aprendida sobre os dados, assim, a partir do conjunto dos descritores de todos os documentos do *dataset*, calcula-se essa transformação uma única vez. Como exposto anteriormente, essa operação descorrelaciona as 1024 dimensões e equaliza suas variâncias, impedindo que poucos canais dominantes ofusquem os demais na comparação, e é apontada como decisiva para o desempenho do R-MAC (Tolias; Sircé; Jégou, 2016). Aprendida uma vez, é então armazenada e reaplicada a cada novo documento submetido em tempo de classificação, de modo que todos residam no mesmo espaço vetorial

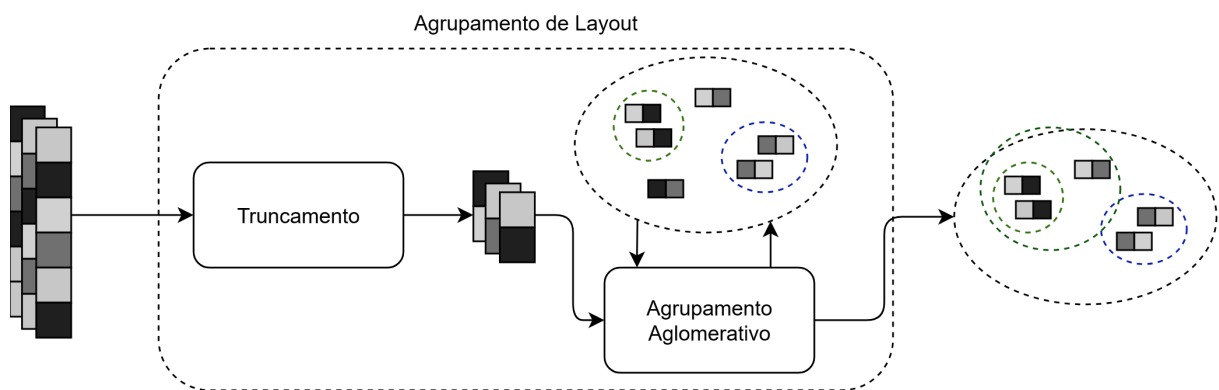
e permaneçam comparáveis.

Vale notar, por fim, que é sobre essa mesma representação que se apoiou a deduplicação realizada na construção do *dataset*, em que cópias exatas de um mesmo documento produzem descritores idênticos e, portanto, têm distância nula entre si no espaço vetorial, o que permitiu detectá-las e removê-las automaticamente. Mais do que limpar o *dataset*, esse comportamento confirma a fidelidade do descritor de *layout*, que sempre atribui a um mesmo documento a mesma posição no espaço vetorial.

5.3.3 Agrupamento de Layout

O processamento realizado por este módulo inicia com a redução de dimensões dos descritores gerados pelo módulo de extração de *layout*. O PCA-whitening, aplicado anteriormente, já dispõe as 1024 dimensões em ordem decrescente de variância, ou seja, das que mais distinguem um documento às que menos distinguem. No entanto, a transformação também reescala as dimensões de baixa variância ao mesmo peso das principais, de modo que possam injetar ruído no espaço vetorial durante as comparações. Dessa forma, mantêm-se apenas os 128 primeiros elementos do vetor, que nesse caso correspondem aos eixos que realmente importam para a paridade, ou disparidade, dos documentos.

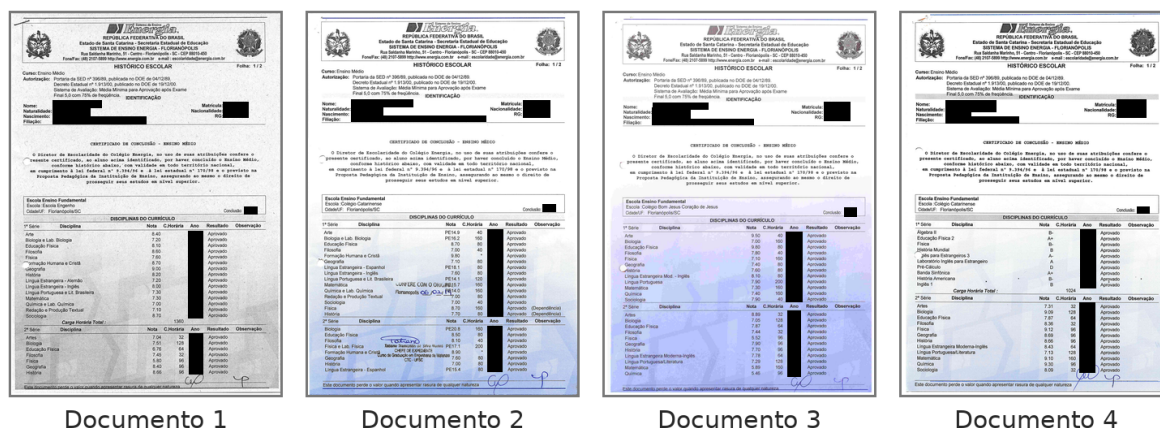
Figura 32 – Detalhamento do Módulo de Agrupamento de Layout



Fonte: o autor

Sobre esses vetores reduzidos aplica-se uma clusterização aglomerativa com *linkage* completo, tomando a distância de cosseno como métrica e 0,55 como limiar de corte, isto é, todo par de um mesmo grupo fica a menos de 0,55 de distância de cosseno entre si, formando grupos enxutos, cada um reunindo instâncias muito similares de um *template*, dessa forma, documentos que não se associam a nenhum outro são considerados como ruído e descartados.

Figura 33 – Amostras de Documentos do Mesmo Cluster de Primeira Ordem



Fonte: o autor

Por reunir apenas documentos mutuamente próximos, essa primeira aglomeração é conservadora, ao passo que mantém separadas as variações de um mesmo modelo que diferem por geração de impressão ou técnica de digitalização (foto, *scanner* etc.). Um segundo estágio de clusterização reagrupa os próprios *clusters* para reconstituir essas famílias de documentos. Cada grupo é representado por dois pontos, seu centroide (a posição média de seus membros) e seu medoide (o membro real mais central, que melhor o representa), e uma nova rodada de agrupamento une aqueles que estão próximos: medoides a até 0,50 ou centroides a até 0,45 de distância. Diferentemente do primeiro estágio, dessa vez o encadeamento transitivo é bem-vindo, já que é ele que reconecta diferentes capturas ou gerações distantes de um mesmo modelo.

Figura 34 – Amostras de Documentos do Mesmo Cluster de Segunda Ordem



Fonte: o autor

A combinação de parâmetros encontrados não foi arbitrária, mas decorre de uma busca experimental. Avaliaram-se métricas de coesão entre *clusters*, como a distância de cosseno entre seus integrantes, em conjunto com a validação visual da coerência dos

agrupamentos gerados a cada ajuste. Assim, uma nova rodada de avaliação era iniciada conforme variavam-se valores de limiar de corte, comprimento do truncamento de dimensões e critérios de *linkage*. Mais detalhes sobre os resultados de cada iteração podem ser encontrados no próximo capítulo, na Tabela 1.

Ao final do processo, cada documento é rotulado com os identificadores dos *clusters* aos quais pertencem, para os dois estágios. Assim, são os *clusters* de segunda ordem, ou seja, as famílias dos documentos, que orientam a geração de falsificações, descrita a seguir, enquanto os documentos deixados como ruído, por não pertencerem a nenhuma família conhecida, são utilizados para a calibragem do *gate* da inferência.

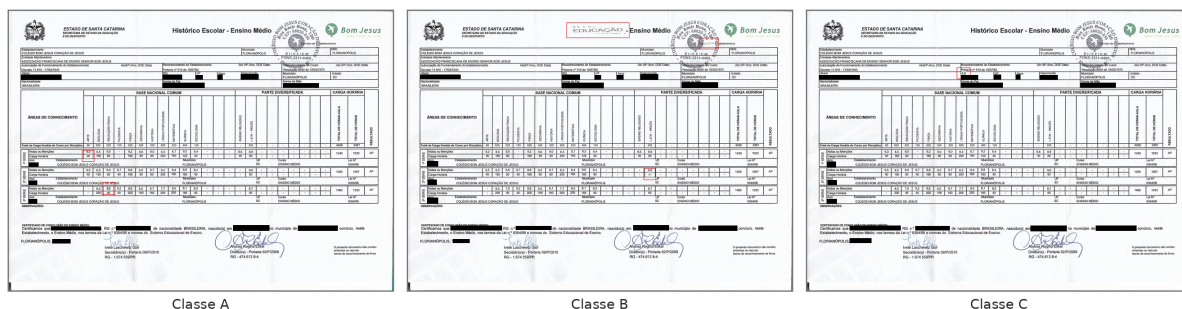
5.3.4 Geração de Falsificações

Para aprender a localizar adulterações, o detector forense precisa de exemplos rotulados, isto é, documentos falsificados acompanhados da indicação exata de quais *pixels* foram alterados. Como presume-se que o *dataset* não contém falsificações, todo esse material de treino precisa ser gerado sinteticamente. Assim, utiliza-se o produto do módulo anterior para que as famílias de *template* forneçam conjuntos de documentos compatíveis entre si, a partir dos quais são forjadas adulterações de vários tipos.

Para reproduzir as falsificações, localizam-se primeiro os campos de texto de cada documento por meio de OCR. Emprega-se a biblioteca EasyOCR, escolhida por oferecer um modelo nativo de português e por fornecer, para cada linha detectada, a caixa delimitadora do texto reconhecido. Além disso, o reconhecimento produz também um grau de confiança, em que adotou-se um limiar permissivo de 0,3, que descarta apenas as detecções mais frágeis, de modo a não perder texto legítimo em cabeçalhos desbotados ou digitalizações ruidosas. Assim, as caixas resultantes indicam à etapa de síntese as regiões de texto sobre as quais uma falsificação pode ser aplicada. A falsificação consiste, então, em editar um ou mais desses campos, registrando as regiões editadas como rótulos que demarcam uma adulteração.

A partir desse mecanismo, definem-se três classes de falsificação com papéis distintos. As classes A e B são variações geradas para ampliar a diversidade do conjunto de treinamento e levar o detector a generalizar o aprendizado, ao passo em que a classe C reproduz o ataque tal como se presume que ocorre na prática. Amostras desses diferentes tipos de falsificação são exibidas na Figura 35:

Figura 35 – Amostras de Documentos Sinteticamente Falsificados



Fonte: o autor

1. Classe A: substitui um campo por outro, retirado de um documento da mesma família e compatível em posição e tamanho. Como ambos os documentos compartilham modelo, fonte e outros aspectos de *layout*, a troca resulta em uma falsificação coerente e difícil de notar;
2. Classe B: similar à classe A, no entanto, o campo substituto tem origem em uma família distinta, de forma que provoque uma incompatibilidade visível de tipografia e renderização. Esse sinal mais grosseiro tem por objetivo reforçar o aprendizado para o reconhecimento de uma quebra de contexto;
3. Classe C: edita o próprio documento por meio da substituição do conteúdo textual de um campo, da cópia de um trecho para outra posição, do apagamento de um campo por preenchimento com o fundo ou de sua cobertura por um retalho neutro.

Cada documento real dá origem a cerca de cinco falsificações, de forma que cada arquivo possua três informações: a imagem falsificada, sua máscara de adulteração e os metadados de proveniência. A partir disso forma-se o *dataset* sintético que treina e avalia o detector.

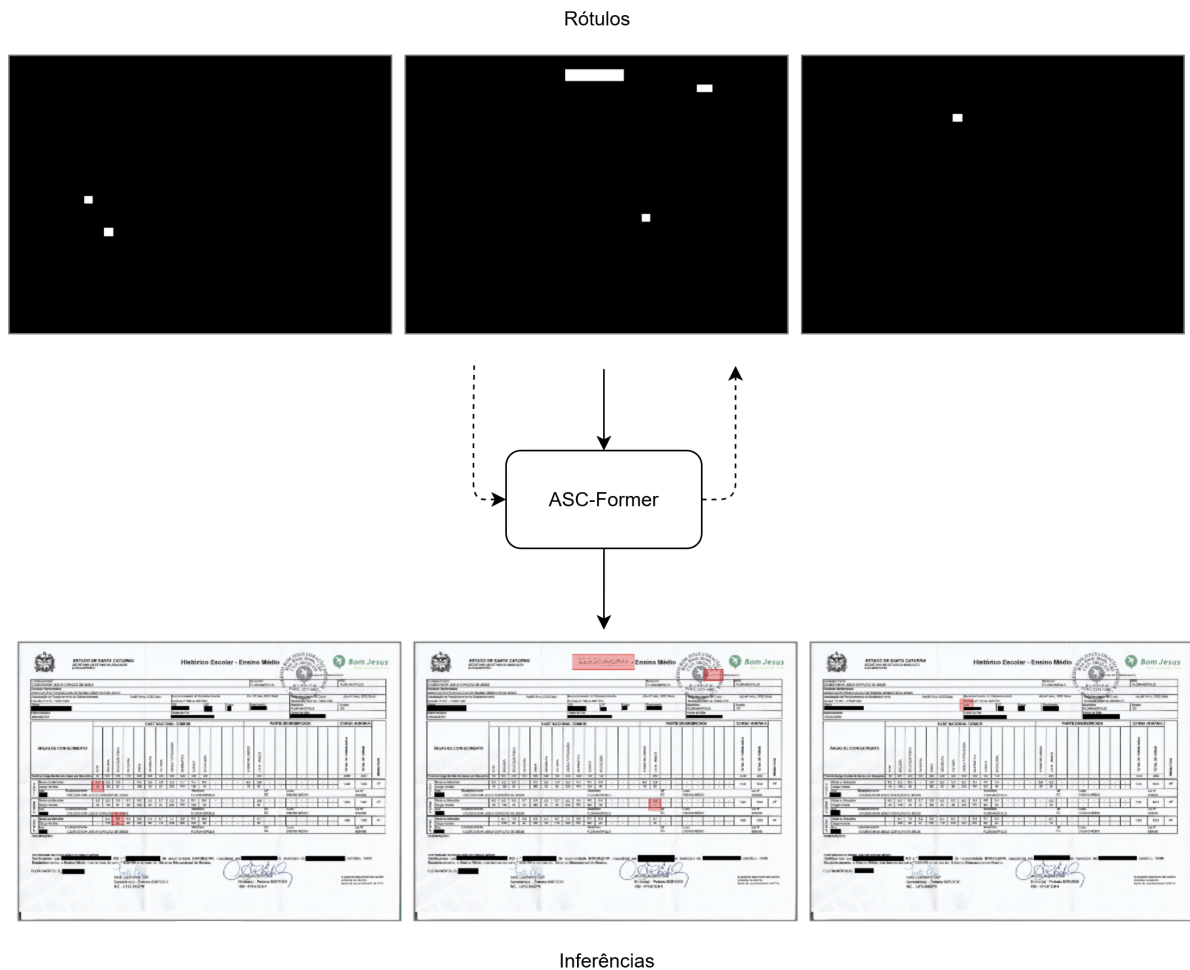
Esse *dataset* é então dividido em treino, validação e teste, na proporção aproximada, respectivamente, de 80%, 10% e 10%, sob restrição que impede qualquer vazamento entre os conjuntos. Cada família de *template* é mantida inteira em um único conjunto, de modo que nenhum modelo visto no treino reapareça na validação ou teste. Além disso, a divisão é feita por documento de origem, e não por falsificação, garantindo outra camada de proteção de que um mesmo documento não apareça em mais de um conjunto, editado de formas diferentes. Dessa forma, o detector é treinado, avaliado e testado com as três diferentes classes de falsificações.

5.3.5 Treinamento do Detector Forense

O detector forense, núcleo de detecção da solução, é o ASC-Former, apresentado no capítulo anterior. Ao invés de treiná-lo do zero, parte-se dos pesos públicos disponibilizados por seus autores, já treinados sobre o conjunto RTM, com mais de quatorze mil imagens de documentos genéricos e recibos, autênticos e adulterados. Como o RTM trata exatamente da mesma tarefa, isto é, a detecção de adulteração de texto em documentos, os pesos resultantes desse treino transferem bem para o caso de uso deste trabalho, e o que resta, então, é um *fine-tuning* que, diferentemente do codificador de *layout* (mantido congelado), retreina todas as camadas do modelo para adaptá-lo aos documentos acadêmicos obtidos e às falsificações produzidas.

O *fine-tuning* é feito sobre todo o *dataset* gerado pelo módulo anterior, com as três classes de falsificação, ao qual se somam os documentos reais autênticos. Estes são indispensáveis, pois cada documento real do conjunto de treino entra também sem nenhuma edição, com uma máscara vazia, o que ensina o modelo a reconhecer um documento íntegro e a não disparar sobre páginas legítimas. Dessa forma, cada documento real contribui com sua versão íntegra e suas cinco falsificações, na proporção de um autêntico para cinco adulterados.

Figura 36 – Ilustração do Treinamento do Detector Forense



Fonte: o autor

Uma página a 200 DPI é custosa demais para ser processada de uma vez durante o treino, de modo que cada passo opera sobre um recorte de 512×512 *pixels*. Como as regiões adulteradas ocupam apenas uma pequena fração de cada documento, metade desses recortes é deliberadamente centrada sobre a máscara de adulteração, garantindo que o modelo veja exemplos de regiões manipuladas com frequência suficiente para aprender a reconhecê-las. A outra metade é sorteada em posições aleatórias, assim como a totalidade dos recortes extraídos dos documentos autênticos. Outra escolha relevante diz respeito à aumento de dados, técnica comum no treino de redes visuais que consiste em aplicar pequenas perturbações às imagens a cada passo, como rotações, ruído ou variações de cor, com o fim de ampliar artificialmente a variedade do conjunto para tornar o aprendizado mais robusto. Neste trabalho, porém, esse recurso é usado com parcimônia, pois muitas dessas perturbações destruiriam justamente os vestígios que se deseja detectar. Além do recorte, a única aumento aplicada às imagens é o espelhamento horizontal aleatório. Essa restrição, contudo, não possui custo prático, visto que a variedade que o aumento

ofereceria já vem da geração das três classes de falsificação.

O procedimento de *fine-tuning* em si reproduz o método do artigo de Luo et al. (2024), em vez de buscar uma configuração própria. Conserva-se, assim, a função de perda original apresentada no capítulo anterior, que consiste da soma de entropia cruzada calculada por *pixel* e o termo contrastivo do módulo TAC com peso 0,05. Mantêm-se também o otimizador original (AdamW) e a taxa de aprendizado que decresce de 6×10^{-5} a zero ao longo de 80 mil iterações. Essa escolha tem por base a premissa de que a arquitetura foi projetada e validada sob esses hiperparâmetros, de modo que empregá-la nas condições para as quais foi concebida reduz o risco de um ajuste malsucedido. Por fim, é com esse arcabouço que produziram-se os pesos ajustados que conformam o detector.

Para converter a saída do detector – um mapa que estima a probabilidade de manipulação de cada região – em uma decisão por documento, ela é condensada em um escore. Como a adulteração afeta apenas uma pequena fração da página, toma-se a média das regiões de maior resposta do mapa, e não de sua totalidade, isto é, ordenam-se os *pixels* pela probabilidade atribuída a cada um e toma-se a média dos k de maior valor, neste caso, com k correspondendo a 0,1% dos *pixels* da página. Essa agregação é conhecida como *mean top-k*, cujo resultado é elevado quando há uma região concentrada de forte resposta, e baixo quando nenhuma se destaca. Evitam-se, assim, tanto a média sobre a imagem inteira, que diluiria um foco pequeno em meio à área autêntica, quanto o valor máximo isolado, sensível demais a um único *pixel* ruidoso.

Assim, para estabelecer a partir de que valor desse score um documento deve ser considerado suspeito, é realizada uma calibragem sobre o conjunto de validação, reservado do treino para esse fim. Aplica-se o detector a seus documentos e toma-se, sobre a distribuição dos escores obtidos, o ponto que melhor distingue as duas classes, isto é, aquele em que a soma entre a taxa de falsificações corretamente detectadas e a de documentos legítimos corretamente aceitos é máxima, critério conhecido como estatística J de Youden. É esse limiar que rege a decisão na etapa de inferência, fixado uma única vez ao fim do treino.

5.4 INFERÊNCIA

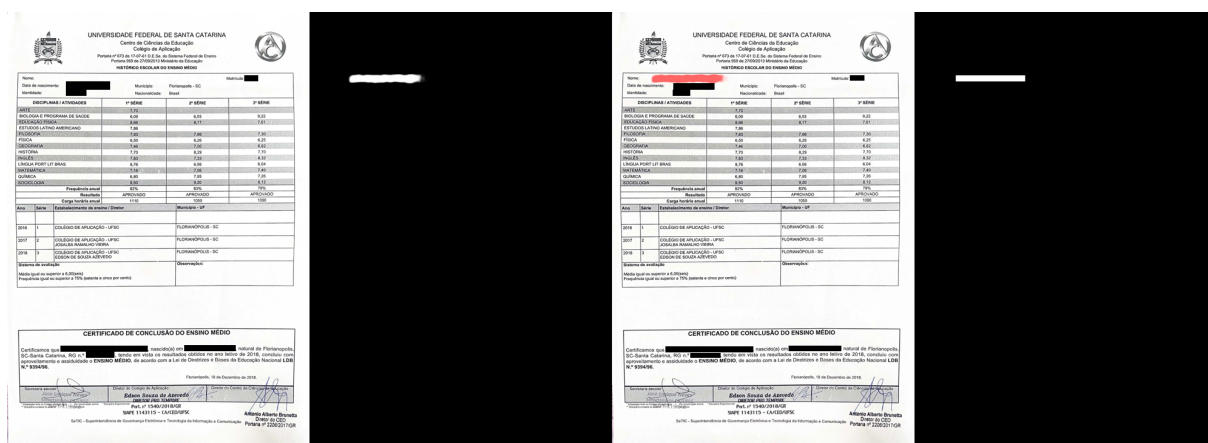
No fluxo de inferência, previamente ilustrado na Figura 27, um documento submetido passa primeiro pela extração de *layout* e, caso pertença a uma família de documentos conhecida, segue à varredura forense, da qual se obtém a avaliação final.

A extração de *layout* cumpre, nesse fluxo, o papel de delimitar a competência do sistema. O descritor R-MAC do documento passa pelo mesmo branqueamento PCA e truncamento ajustados no treino e é atribuído, por proximidade, à família de *template* mais próxima, medida pela distância de cosseno a seu medoide, desde que abaixo do limiar que originou as famílias. Quando o descritor fica distante de todas, dispara-se o *gate* de pertencimento e o documento é tido como de modelo desconhecido. Esse reconhecimento é

condição necessária para a etapa seguinte, pois diante de um modelo que o sistema nunca viu, não há garantia sobre quais adulterações seriam plausíveis nem base de calibração para avaliá-las.

Reconhecida a família, o documento segue para o ramo forense, em que o detector varre a imagem por inteiro e devolve o mapa de adulteração em nível de *pixel*, condensado no escore e comparado ao limiar de decisão fixado no treino. Classifica-se o documento como suspeito quando o escore ultrapassa o limiar, e como normal caso contrário, sempre acompanhado do mapa de adulteração, que evidencia as regiões responsáveis pela decisão e confere explicabilidade ao resultado.

Figura 37 – Exemplo de Resultado de Inferência (no painel: o documento falsificado anonimizado; o mapa de adulteração; overlay do mapa de adulteração no documento; o rótulo de adulteração)



Fonte: o autor

Cabe registrar que esse encadeamento expõe uma limitação de arquitetura. Por condicionar a detecção forense ao reconhecimento prévio do *layout*, o sistema avalia apenas documentos cujos *templates* constam em sua base de dados de referência, e aponta como desconhecidos aqueles que não se associam a nenhuma família conhecida, que podem então passar por um processo de revisão manual por um especialista do domínio, aferindo sua autenticidade pelos meios convencionais de verificação. Essa restrição, porém, preserva uma propriedade desejável, pois a solução não está atrelada a tipos específicos de documento, mas ao modelo de ameaça. À medida que os documentos desconhecidos se acumulam e são confirmados de se enquadrarem nesse modelo, uma nova rodada de treinamento pode incorporar suas famílias, estendendo o alcance do sistema sem alterar o método.

6 RESULTADOS

Este capítulo apresenta os resultados e métricas obtidos em cada etapa da solução.

6.1 CONFIGURAÇÃO EXPERIMENTAL E DATASET

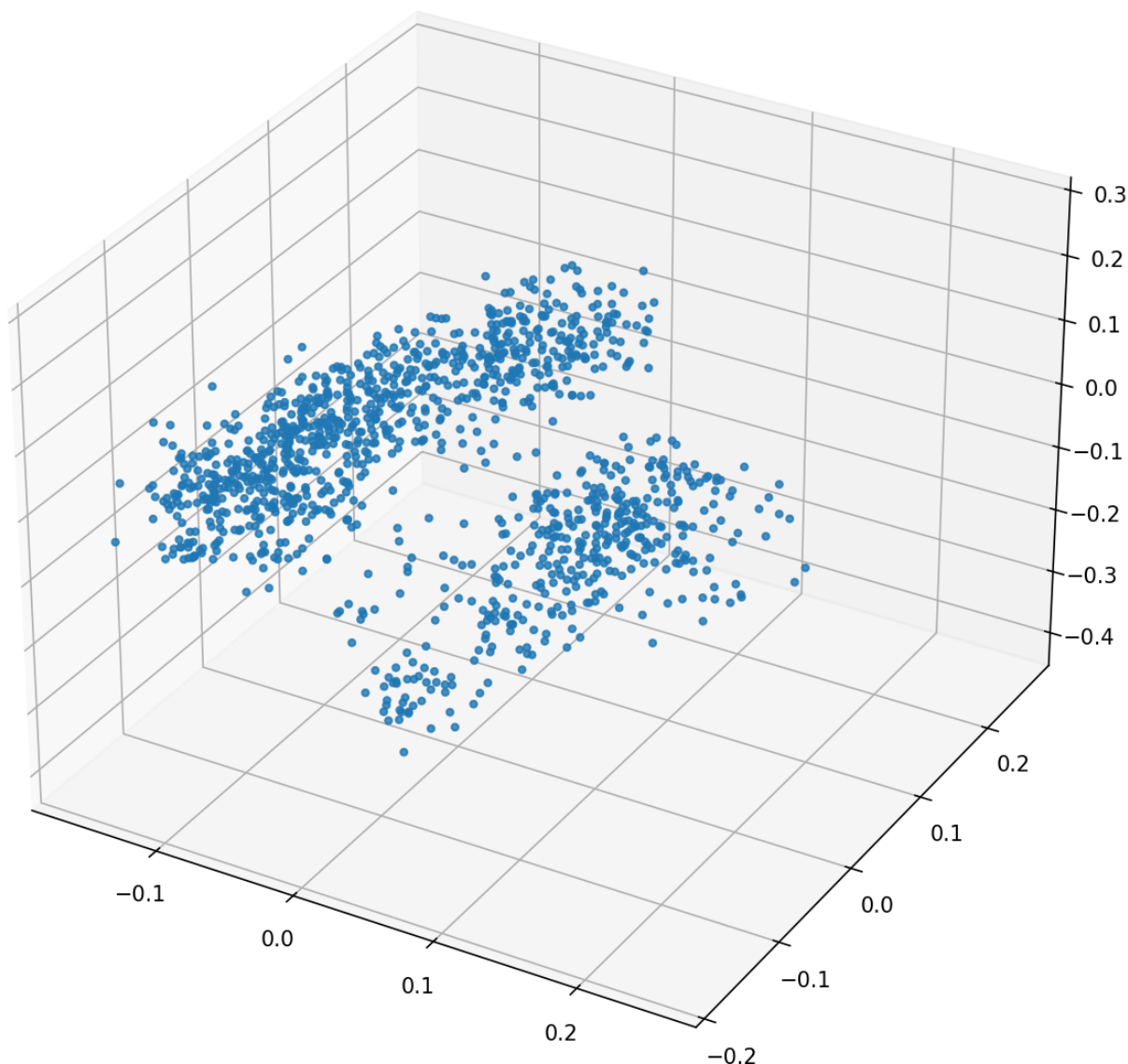
Como mencionado no Capítulo 5, o *dataset* tem origem em 2161 arquivos XML no esquema de diploma digital do MEC, dos quais 2066 traziam documentos relevantes. Rasterizados página a página a 200 DPI, esses documentos produziram 2105 imagens, número reduzido a 1989 após a deduplicação, conjunto sobre o qual todo o treinamento e avaliação foram conduzidos.

Os experimentos correram em dois ambientes distintos. Executaram localmente, em uma GPU AMD RX 9070 XT sob ROCm 7.2, a extração e o pré-processamento dos documentos, a extração e o agrupamento de *layout* e a geração de falsificações, que, para a iteração final, somaram cerca de 12 horas. Desse total, aproximadamente 6 horas couberam à extração e ao pré-processamento, etapa em que o OCR correspondeu quase exclusivamente ao total do tempo; 4 horas à extração e ao agrupamento de *layout*, que couberam em maioria à codificação via DINOv2; e 2 horas à geração de falsificações. O *fine-tuning* do detector forense, computacionalmente mais exigente, correu no Laboratório Virtual Multiusuário de Ensino (VLAB) da UFSC, em duas GPUs NVIDIA H100 sob CUDA 12.4 e com PyTorch 2.5.1, totalizando cerca de 33 horas para as 80 mil iterações. Para assegurar a reprodutibilidade durante as iterações, fixou-se uma semente mestra em todas as etapas estocásticas.

6.2 DESCOBERTA DE TEMPLATES

O agrupamento de *layout* sobre os 1989 documentos produziu 473 *clusters* no primeiro estágio, deixando 757 documentos (38,1%) como ruído, isto é, sem vizinhos próximos o suficiente para formar um grupo. Os *clusters* resultantes são pequenos e coesos, com mediana de 2 e máximo de 14 documentos, e distância média de cada membro ao medoide de seu grupo de apenas 0,139, o que indica grande homogeneidade interna.

Figura 38 – Representação dos Clusters de Primeira Ordem



Fonte: o autor

A escolha do critério de *linkage* e do limiar de corte do primeiro estágio apoiaram-se na varredura da Tabela 1. Cada linha corresponde a uma configuração de *linkage* e de limiar de corte, e as três colunas seguintes registram a distância máxima intragrupo, o número de grupos formados e a fração de documentos deixados sem grupo, ou ruído. A distância máxima intragrupo, definida como a maior distância de cosseno entre dois documentos de um mesmo grupo, revela a diferença essencial entre os critérios, pois o *linkage* completo mantém todo par abaixo do limiar de corte, enquanto o de média, por restringir apenas a distância média entre os pares, admite no mesmo grupo documentos a até 0,940 de distância, mais do que separa documentos sem relação alguma. Embora o *linkage* de média produza menos grupos e menos ruído, esses números enganam, pois resultam da absorção de documentos heterogêneos em um mesmo *cluster*. Adotou-se, portanto, o *linkage* completo

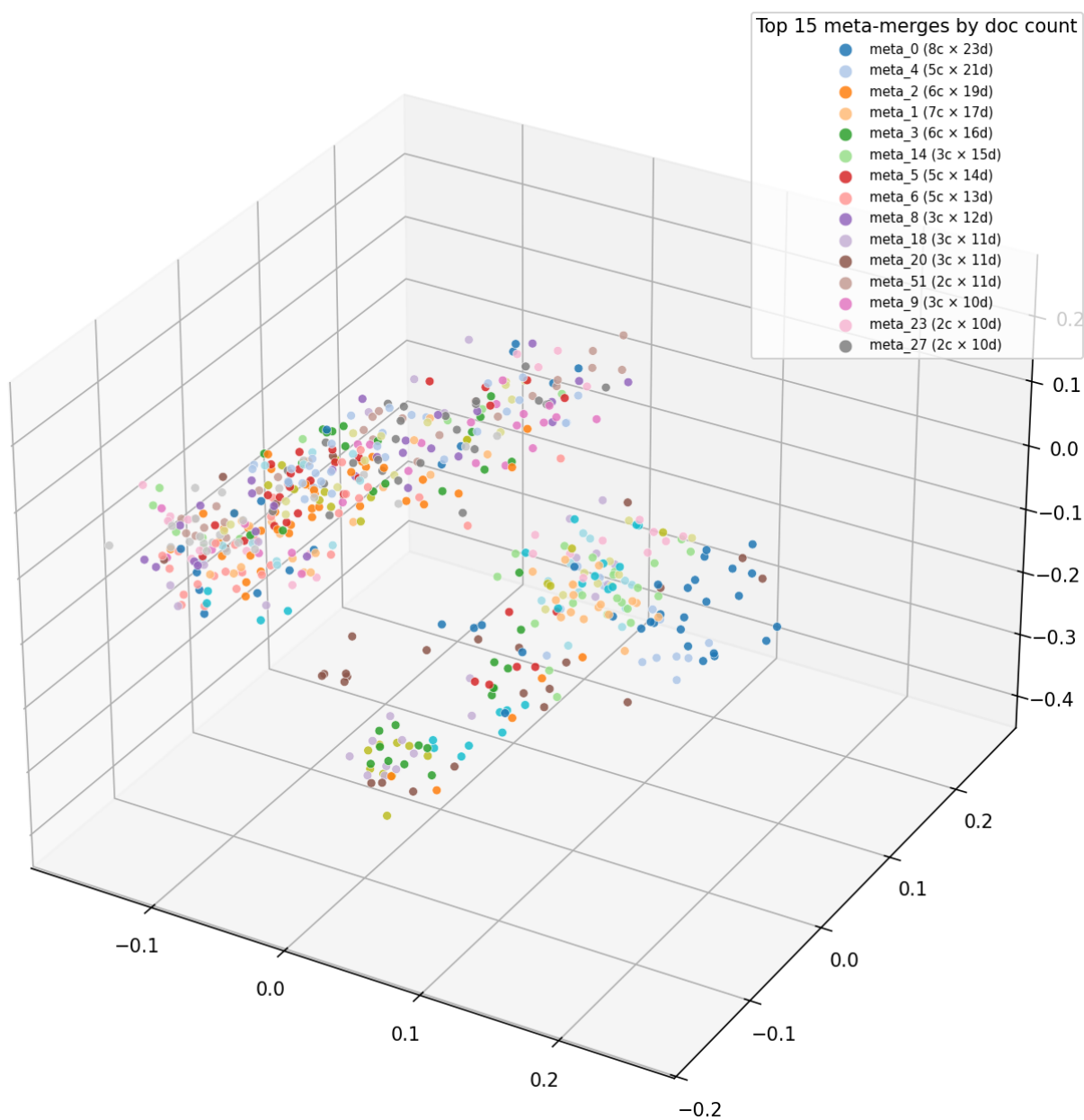
e, entre seus limiares, o valor 0,55, o mais estrito dos avaliados, que elimina misturas residuais ainda toleradas por limiares mais frouxos e produz os 473 grupos e os 38,1% de ruído já reportados.

Tabela 1 – Seleção do *linkage* e do limiar de corte do primeiro estágio

<i>Linkage</i>	Limiar de corte	Dist. máx. intragrupo	Clusters	Ruído
Completo	0,55	0,550	473	38,1%
Completo	0,60	0,599	525	26,2%
Completo	0,65	0,650	560	16,6%
Média	0,55	0,871	391	36,9%
Média	0,60	0,871	432	24,9%
Média	0,65	0,940	438	15,3%

A fração elevada de ruído e o tamanho reduzido dos grupos refletem uma característica conhecida do problema, em que um mesmo modelo de documento aparece sob diferentes gerações de impressão e digitalização que o codificador trata como distintas. Para reconciliar esses fragmentos, o segundo estágio de agrupamento formou 83 famílias de *template* ao reunir grupos dispersos do primeiro estágio, abrangendo 596 documentos, 48,4% dos que haviam sido agrupados, enquanto 265 *clusters* permaneceram isolados.

Figura 39 – Representação dos Clusters de Segunda Ordem



Fonte: o autor

Além das métricas quantitativas, a validação do agrupamento foi conduzida de forma qualitativa, em que adotou-se como critério de aceitação que ao menos os vinte maiores *clusters* fossem visualmente homogêneos quanto ao modelo de documento. A condição foi satisfeita e constatou-se que as famílias formadas no segundo estágio reúnem, de fato, variações de um mesmo *template*, ainda que capturadas em diferentes gerações de impressão e digitalização.

6.3 GERAÇÃO DE FALSIFICAÇÕES

A geração de falsificações produziu 16093 amostras, cada qual composta pela imagem adulterada, sua máscara de adulteração e os metadados da edição, ocupando cerca de 41 GB. As amostras distribuem-se nas três classes descritas no Capítulo 5: 3665 da classe A (troca de campo dentro da mesma família), 6192 da classe B (troca entre famílias distintas) e 6236 da classe C (adulterações por cópia, preenchimento, cobertura e inserção dentro do próprio documento).

As manipulações tem caráter pontual, assim, a máscara de adulteração cobre, em média, pouco mais de 1% da página nas classes A e B e apenas 0,2% na classe C. Nas classes de troca de campo, cada falsificação altera, em média, 4,4 campos. A taxa de compatibilidade geométrica entre os campos trocados foi de 84,7% na classe A e de 67,5% na classe B, diferença esperada, já que trocas dentro de uma mesma família encontram campos de posição e tamanho semelhantes com mais facilidade do que trocas entre famílias distintas. Dessa forma, as falsificações variam bastante em perceptibilidade. A Figura 40 ilustra dois extremos, opondo uma falsificação da classe C, cuja cópia de um trecho do próprio documento reproduz o conteúdo legítimo deixando marcas pouco visíveis, a uma da classe B, em que o campo trazido de uma família distinta pode ser percebido imediatamente.

suas máscaras.

Tabela 2 – Estratificação em treino, validação e teste

Conjunto	Documentos reais	Falsificações
Treino	1.590	12.553
Validação	200	1.582
Teste	199	1.958
Total	1.989	16.093

6.4 DETECTOR FORENSE SUPERVISIONADO

Para medir a eficácia do detector, recorre-se a um conjunto de métricas padrão de classificação:

- Precisão: é a fração, dentre os documentos apontados como suspeitos, que de fato é falsificada;
- *Recall*: é a fração das falsificações existentes que é efetivamente detectada;
- F1: é a média harmônica entre precisão e *recall*, penalizando o desequilíbrio entre os dois;
- Área Sob a Curva de Característica de Operação do Receptor (AUROC): mede a separação entre falsificações e autênticos independentemente do limiar, de 0, 5, em uma decisão aleatória, a 1, 0, na separação perfeita;
- Interseção Sobre União (IoU): mede em nível de *pixel* a sobreposição entre a região apontada como adulterada e a verdadeira, dada pela razão entre a área comum às duas e a área total que cobrem.

O detector desenvolvido opera com a agregação e o limiar fixados no treino, isto é, a *mean top-k* com corte em 0,3565 sobre a imagem inteira em resolução plena. A Tabela 3 apresenta seu desempenho em nível de documento para cada classe de falsificação, em que as métricas foram tomadas sobre o conjunto de teste. A métrica central deste trabalho é o F1 da classe C, que reproduz o modelo de ameaça mais realista, e alcança um ótimo valor de 0,850. As classes A e B, de troca de campo, são mais fáceis de detectar e alcançam valores superiores. Vale notar que a classe C combina precisão muito alta com recall mais modesto, o que revela um detector conservador, que raramente acusa em falso, mas ainda deixa escapar cerca de um quarto das falsificações.

Tabela 3 – Desempenho do detector a nível de documento, por classe

Classe	F1	Precisão	Recall	AUROC
A	0,958	0,957	0,959	0,985
B	0,973	0,974	0,973	0,993
C	0,850	0,978	0,751	0,900

A maior dificuldade da classe C decorre da natureza de suas manipulações, que são mais sutis e localizadas que as trocas de campo. Ainda assim, quando a falsificação é detectada, sua localização permanece precisa, como mostra a Tabela 4, com AUROC de *pixel* acima de 0,986 nas três classes. A taxa de falsos positivos foi de 8,0% no total, isto é, sobre os 199, 16 documentos legítimos foram sinalizados como suspeito.

Tabela 4 – Localização do detector a nível de *pixel*, por classe

Classe	F1	IoU	AUROC
A	0,873	0,809	0,996
B	0,922	0,868	1,000
C	0,701	0,640	0,986

Para complementar as métricas agregadas, conduziu-se uma inspeção visual do detector sobre 25 amostras de cada tipo, totalizando 100 amostras do conjunto de teste, em que cada documento foi examinado por um painel com a imagem original, o mapa de calor de adulteração, sua sobreposição e a máscara de rótulo verdadeira. A Tabela 5 resume essa vistoria, registrando, além da contagem de documentos sinalizados, o escore médio atribuído a cada tipo. Os escores separam com folga falsificações e autênticos, de 0,938 na Classe A a 0,638 na Classe C, a mais difícil, contra apenas 0,125 nos documentos autênticos, o que confirma visualmente o comportamento observado nas métricas e sustenta a robustez do limiar de decisão.

Tabela 5 – Resumo da inspeção visual do detector, com 25 amostras por tipo, indicando os documentos sinalizados e o escore médio

Tipo	Sinalizados	Escore médio
A	24	0,938
B	24	0,888
C	18	0,638
Autêntico	2	0,125

6.5 LIMITAÇÕES OBSERVADAS

A primeira limitação pode ser notada no agrupamento de *layout*. Embora a inspeção visual confirme que os grupos formados são puros, uma mesma família de *template* acaba repartida entre vários *clusters* de primeira e segunda ordem, o que infla o número de grupos resultantes, com 83 famílias e 265 *clusters* ainda isolados, além do número expressivo de 38,1% de documentos tratados como ruído. Como o *dataset* reúne um número estimado de poucas dezenas de modelos de documento, esperaria-se menos de uma centena de grupos ao todo.

Depois, referente à detecção forense, ao investigar por que a classe C apresentou métricas inferiores às demais, notou-se que a sensibilidade do detector varia conforme a natureza e o tamanho da adulteração. Dentro dessa classe, as falsificações capturadas

com maior confiança são os apagamentos e preenchimentos de campos por cor neutra, que cobrem áreas extensas e uniformes; e as substituições de conteúdo textual, em que a fonte empregada na falsificação destoa com a do documento original e torna a edição notável. Em contrapartida, a sensibilidade cai quando a adulteração se restringe a campos muito pequenos, como a troca de um dígito de uma nota, cuja área diminuta não produz vestígios grandes o suficiente para que o detector capture com segurança. Essa fragilidade é especialmente preocupante porque o modelo de ameaça previsto contempla justamente essa alteração pontual, em que adulterar um único dígito pode bastar para forjar um histórico.

Um impacto direto dessa calibragem entre confiança e sinalização é notada na taxa de falsos positivos de 8,0% sobre documentos autênticos. Um limiar mais alto reduziria os falsos positivos a menos de 1%, mas ao custo de deixar passar ainda mais falsificações da classe C. O limiar adotado é um compromisso entre detecção e confiança, aceitando sinalizar cerca de um documento legítimo em cada treze.

Além disso, também foi notado que os falsos positivos não se distribuem de modo uniforme, mas concentram-se em documentos com ruído de captura adicional, como digitalizações de baixa qualidade ou redigitalizadas, em que o ruído de renderização da própria captura se assemelha aos vestígios que o detector associa à adulteração.

Por fim, cabe registrar uma limitação de escopo. Todo o sistema foi desenvolvido e avaliado sobre um único domínio, o de documentos acadêmicos brasileiros de um mesmo *dataset*, de modo que as famílias de *template* descobertas, a calibragem do *gate* e os vestígios de adulteração aprendidos pelo detector são específicos a esse contexto. Não se verificou como a solução se comporta diante de outros tipos de documento, de instituições fora do *dataset* ou de estilos de falsificação distintos dos sintetizados, e sua validação empírica permanece, portanto, circunscrita a esse domínio.

7 CONCLUSÃO

Este trabalho apresentou a construção de um protótipo de software capaz de classificar documentos acadêmicos quanto à suspeita de fraude. Propõe-se que essa classificação automatizada sirva de apoio a um processo conduzido por um especialista, atuando como ferramenta auxiliar à decisão, e não como veredito final. O *pipeline* desenvolvido, que encadeia o agrupamento de diferentes *templates* de documentos, a geração de falsificações sintéticas e o detector forense, mostrou-se eficaz, alcançando F1 de 0,850 na classe de falsificações mais realistas e valores ainda superiores nas demais, ainda que se identifique como principal limitação a menor sensibilidade a adulterações muito pequenas. Como o detector especializa, para o domínio acadêmico aqui tratado, a arquitetura ASC-Former pré-treinada sobre o *dataset* RTM, o F1 de 0,807 que o artigo original reporta sobre esse *dataset* (Luo et al., 2024) serve de referência, e o F1 de 0,850 obtido mostra-se compatível com tal patamar, indício de que a especialização transferiu bem para o caso de uso. Não se trata, porém, de superar o modelo original, dado que os *datasets* e as classes de falsificação avaliadas diferem.

A solução foi concebida centrada no modelo de ameaça, e portanto desacoplada de um tipo específico de documento. Isso possibilita ao sistema operar sobre quaisquer famílias que se enquadrem nesse modelo, já que pode ser periodicamente retreinado para incorporar tipos de *layout* que não reconhece, à medida que se acumulam. Além disso, a arquitetura modular admite a inclusão de novos módulos sempre que um novo requisito ou caso de uso os exigir.

Quanto a perspectivas futuras, alguns caminhos se destacam. Para mitigar a fraqueza diante de adulterações minúsculas, vale investigar o treinamento com maior diversidade de manipulações pontuais e a adoção de limiares de decisão por família de *template*. Além disso, a robustez a documentos fotografados ou redigitalizados pode ser ampliada com aumento específica para esse tipo de ruído. Também pode ser explorado um *fine-tuning* contrastivo do codificador para o agrupamento de *layout*, que permitiria reconciliar as gerações de impressão e digitalização que hoje fragmentam as famílias. Por fim, a integração com a Jornada do Estudante levaria a solução ao ambiente para o qual foi concebida.

REFERÊNCIAS

- AGRAWAL, Shikha; AGRAWAL, Jitendra. Survey on Anomaly Detection using Data Mining Techniques. **Procedia Computer Science**, v. 60, p. 708–713, 2015. ISSN 1877-0509. DOI: 10.1016/j.procs.2015.08.220.
- ALAMERI, Mohammed et al. Unsupervised Forgery Detection of Documents: A Network-Inspired Approach. **Electronics**, v. 12, abr. 2023. DOI: 10.3390/electronics12071682.
- ALZUBAIDI, Laith et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. **Journal of Big Data**, v. 8, n. 1, p. 53, mar. 2021. ISSN 2196-1115. DOI: 10.1186/s40537-021-00444-8. Acesso em: 27 jun. 2025.
- BELLMAN, Richard E. **Dynamic Programming**. Princeton, NJ, USA: Princeton University Press, 1957. (Rand Corporation research study). ISBN 9780691079516.
- BENGIO, Yoshua; COURVILLE, Aaron; VINCENT, Pascal. **Representation Learning: A Review and New Perspectives**. [S.l.: s.n.], 2014. arXiv: 1206.5538 [cs.LG]. Disponível em: <https://arxiv.org/abs/1206.5538>.
- BIRAJDAR, Gajanan K.; MANKAR, Vijay H. Digital image forgery detection using passive techniques: A survey. **Digital Investigation**, v. 10, n. 3, p. 226–245, 2013. DOI: 10.1016/j.diin.2013.04.007.
- BOONKRONG, Sirapat. Design of an academic document forgery detection system. **International Journal of Information Technology**, p. 1–13, jun. 2024. DOI: 10.1007/s41870-024-02006-6.
- BRASIL. **Censo da Educação Superior 2023: notas estatísticas**. Brasília, DF: Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), 2024.
- COZZOLINO, Davide; VERDOLIVA, Luisa. **Noiseprint: a CNN-based camera model fingerprint**. [S.l.: s.n.], 2018. arXiv: 1808.08396 [cs.CV]. Disponível em: <https://arxiv.org/abs/1808.08396>.
- DIAS, Pâmela; LEAL, Arthur. **Sites vendem diploma de curso superior para quem sequer pisou em sala de aula: 'Documentação 100% original, emitida de dentro da universidade', diz atendente**. 2022. Disponível em: <https://oglobo.globo.com/brasil/noticia/2022/11/sites-vendem-diploma-de-curso-superior-para-pessoas-que-nao-concluíram-ou-sequer-pisaram-em-uma-universidade.gh.html>. Acesso em: 5 abr. 2025.
- DIETTERICH, Thomas G. Machine learning. In: **ENCYCLOPEDIA of Computer Science**. GBR: John Wiley e Sons Ltd., 2003. p. 1056–1059. ISBN 0470864125.
- DOSOVITSKIY, Alexey et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2021. DOI: 10.48550/arXiv.2010.11929.

FULLÉR, Róbert; KÁROLY, Artúr István; GALAMBOS, Péter. Unsupervised Clustering for Deep Learning: A tutorial survey. **Acta Polytechnica Hungarica**, v. 15, p. 29–53, dez. 2018. DOI: 10.12700/APH.15.8.2018.8.2.

GLOBO, O. **Quatro pessoas são presas pela venda de 50 mil diplomas falsos e milhares carteirinhas de estudante**. 2025. Disponível em: <https://oglobo.globo.com/brasil/noticia/2022/11/sites-vendem-diploma-de-curso-superior-para-pessoas-que-nao-concluíram-ou-sequer-pisaram-em-uma-universidade.ghml>. Acesso em: 5 abr. 2025.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Acesso em: 16 jun. 2025.

GRIRA, Nizar; CRUCIANU, Michel; BOUJEMAA, Nozha. Unsupervised and Semi-supervised Clustering: a brief survey. **A Review of Machine Learning Techniques for Processing Multimedia Content**, set. 2005.

ISLAM, Noman; ISLAM, Zeeshan; NOOR, Nazia. A Survey on Optical Character Recognition System. **ITB Journal of Information and Communication Technology**, dez. 2016. DOI: 10.48550/arXiv.1710.05703.

JAIN, Rajiv; WIGINGTON, Curtis. Multimodal Document Image Classification. **2019 International Conference on Document Analysis and Recognition (ICDAR)**, p. 71–77, 2019. DOI: 10.1109/ICDAR.2019.00021.

JAISWAL, Ashish; BABU, Ashwin Ramesh et al. **A Survey on Contrastive Self-supervised Learning**. [S.l.: s.n.], 2021. arXiv: 2011.00362 [cs.CV]. Disponível em: <https://arxiv.org/abs/2011.00362>.

JAISWAL, Garima; SHARMA, Arun; YADAV, Sumit. Deep feature extraction for document forgery detection with convolutional autoencoders. **Computers & Electrical Engineering**, v. 99, p. 107770, abr. 2022. DOI: 10.1016/j.compeleceng.2022.107770.

JAMES, Hailey; GUPTA, Otkrist; RAVIV, Dan. OCR Graph Features for Manipulation Detection in Documents, set. 2020. DOI: 10.48550/arXiv.2009.05158.

JOLLIFFE, Ian; CADIMA, Jorge. Principal component analysis: a review and recent developments. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 374, p. 20150202, abr. 2016. DOI: 10.1098/rsta.2015.0202.

KIM, Seong-Kyu. Blockchain Smart Contract to Prevent Forgery of Degree Certificates: Artificial Intelligence Consensus Algorithm. **Electronics**, v. 11, p. 2112, jul. 2022. DOI: 10.3390/electronics11142112.

LUO, Dongliang et al. **Toward Real Text Manipulation Detection: New Dataset and New Solution**. [S.l.: s.n.], 2024. arXiv: 2312.06934 [cs.CV]. Disponível em: <https://arxiv.org/abs/2312.06934>.

MEC. **Aplicativo do MEC ganha prêmio de reconhecimento nacional**. 2022. Disponível em: <https://www.gov.br/mec/pt-br/assuntos/noticias/2022/aplicativo-do-mec-ganha-premio-de-reconhecimento-nacional>. Acesso em: 13 maio 2025.

MEC. **Portaria MEC/DAU n 33 de 2 de agosto de 1978**: Estabelece a sistemática para o registro de diplomas de curso superior. Brasília, DF: Ministério da Educação do Brasil, 1978.

MENGHANI, Gaurav. Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better. **ACM Computing Surveys**, Association for Computing Machinery (ACM), v. 55, n. 12, p. 1–37, mar. 2023. ISSN 1557-7341. DOI: 10.1145/3578938.

MOHAMMED, Shamsudeen; NWOBODO, Lois; EKENE, Njoku. Certificate Fraud Verification Model Using Clustered-Based Classification Approach. **Explorematics Journal of Innovative Engineering and Technology**, v. 5, n. 1, p. 60–72, jun. 2024. ISSN 2636-590. Disponível em: https://www.researchgate.net/publication/381164278_CERTIFICATE_FRAUD_VERIFICATION_MODEL_USING_CLUSTERED-BASED_CLASSIFICATION_APPROACH. Acesso em: 5 jul. 2025.

OQUAB, Maxime et al. **DINOv2: Learning Robust Visual Features without Supervision**. [S.l.: s.n.], 2024. arXiv: 2304.07193 [cs.CV]. Disponível em: <https://arxiv.org/abs/2304.07193>.

PALMA, Lucas M. et al. Blockchain and Smart Contracts for Higher Education Registry in Brazil. **International Journal of Network Management**, v. 29, 2019. DOI: <https://doi.org/10.1002/nem.2061>.

QU, Chenfan et al. Towards Robust Tampered Text Detection in Document Image: New Dataset and New Solution. In: PROCEEDINGS of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2023.

RNP, Rede. **Blockchain da jornada acadêmica**. 2023. Disponível em: <https://www.youtube.com/watch?v=xqezMbjCeTM>. Acesso em: 13 maio 2025.

SARKER, Iqbal. Machine Learning: Algorithms, Real-World Applications and Research Directions. **SN Computer Science**, v. 2, mar. 2021. DOI: 10.1007/s42979-021-00592-x.

SHI, Baoguang; BAI, Xiang; YAO, Cong. An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, PP, jul. 2015. DOI: 10.1109/TPAMI.2016.2646371.

TOLIAS, Giorgos; SICRE, Ronan; JÉGOU, Hervé. **Particular object retrieval with integral max-pooling of CNN activations**. [S.l.: s.n.], 2016. arXiv: 1511.05879 [cs.CV]. Disponível em: <https://arxiv.org/abs/1511.05879>.

VASWANI, Ashish et al. Attention Is All You Need, jun. 2017. DOI: 10.48550/arXiv.1706.03762.

WANG, Haohan; RAJ, Bhiksha. On the Origin of Deep Learning, fev. 2017. DOI: 10.48550/arXiv.1702.07800.

ZHAO, Zengqi et al. **DOCFORGE-BENCH: A Comprehensive Benchmark for Document Forgery Detection and Analysis**. [S.l.: s.n.], mar. 2026. DOI: 10.48550/arXiv.2603.01433.

Apêndices

APÊNDICE A – ACESSO AO CÓDIGO-FONTE

Todo o código desenvolvido neste trabalho junto com suas instruções de uso, encontram-se disponíveis publicamente. O material pode ser acessado no repositório institucional da UFSC, em <https://codigos.ufsc.br/samuel.moreira.ransolin/document-forgery-detection>, ou em seu espelho no GitHub, em <https://github.com/mr-samvel/document-forgery-detection>.

APÊNDICE B – ARTIGO

Reproduz-se, a seguir, na íntegra, o artigo produzido a partir deste trabalho, redigido no formato da Sociedade Brasileira de Computação (SBC).

Detecção de Documentos Acadêmicos Falsificados: Uma Solução Baseada em Aprendizado de Máquina

Samuel M. Ransolin¹, Giovana N. Inocêncio¹, Jean E. Martina¹

¹Departamento de Informática e Estatística – Centro Tecnológico
Universidade Federal do Santa Catarina (UFSC) – Florianópolis, SC – Brasil

samuransolin@gmail.com, gioinocencio017@gmail.com, jean.martina@ufsc.br

Abstract. *In recent years, the significant growth in the number of incoming students, graduates, and higher education institutions in Brazil has brought challenges related to validating the authenticity of academic certificates. Currently, this process relies on predominantly manual verification, which is prone to errors and failures, such as the acceptance of forged documents. In this context emerges the Jornada do Estudante, a system provided by the Ministério da Educação (MEC) that offers, among other features, the tracking of students' academic records through a permissioned distributed network, ensuring that only digitally verified educational institutions can register credits and certifications. This undergraduate thesis revisits the state of the art in the machine-learning-based detection of forged documents and develops a prototype of a hybrid solution that, from the digitized image of academic documents, chains together the clustering of documents into families sharing the same template, the generation of forgeries for training, and a supervised forensic detector capable of localizing tampering at the pixel level, classifying each instance according to its degree of legitimacy. Evaluated on the most realistic class of forgeries, which reproduces subtle, localized tampering in document fields, the detector could identify 75% of the tampered documents, with a precision of 97% among the instances it flagged as fraudulent. By integrating this system into the Jornada do Estudante, documents can be automatically validated before being recorded on the blockchain, significantly increasing the security and reliability of the accreditation process.*

Resumo. *Nos últimos anos, no Brasil, o expressivo aumento do número de ingressantes, formandos e de instituições de ensino superior, trouxe consigo desafios relacionados à validação da autenticidade de certificados acadêmicos. Atualmente, o processo consiste na verificação de forma predominantemente manual, sujeita a erros e falhas, como a aceitação de documentos falsificados. Nesse contexto, surge a Jornada do Estudante, um sistema disponibilizado pelo Ministério da Educação (MEC), que oferece, entre outras funcionalidades, o acompanhamento de registros acadêmicos de estudantes por meio de uma rede distribuída permissionada, de forma a garantir que somente instituições de ensino digitalmente comprovadas possam registrar créditos e certificações. O presente trabalho de conclusão de curso revisita o estado da arte em detecção de documentos falsificados por aprendizado de máquina e desenvolve um protótipo de solução híbrida que, a partir da imagem digitalizada de documentos acadêmicos, encadeia o agrupamento em famílias de mesmo*

template, a geração de falsificações para treinamento e um detector forense supervisionado, capaz de localizar adulterações no nível do pixel, classificando cada exemplar conforme seu grau de legitimidade. Avaliado sobre a classe de falsificações mais realista, que reproduz adulterações sutis e localizadas em campos do documento, o detector conseguiu detectar 75% dos documentos adulterados, com precisão de 97% entre os exemplares apontados como fraudulentos. Ao integrar esse sistema à Jornada do Estudante, é possível validar automaticamente os documentos antes de seu registro em blockchain, aumentando significativamente a segurança e a confiabilidade do processo de credenciamento.

1. Introdução

Ao longo da última década, observa-se no Brasil um crescimento contínuo na emissão de diplomas de ensino superior, com um aumento superior a 31% de formandos desde 2013 [INEP 2024]. Isso traz à tona uma série de desafios a serem superados, entre eles a temática explorada neste estudo: a melhoria nos processos de regulação, supervisão e avaliação dessas emissões por parte do Ministério da Educação do Brasil (MEC).

Atualmente, a gerência, armazenamento e emissão de documentos acadêmicos, como diplomas e históricos escolares, é responsabilidade da instituição de ensino que os emite. O processo, burocrático e não computadorizado, é suscetível a erros humanos em verificações manuais, à ação de funcionários desonestos que podem forjar ou vender certificados, e à perda definitiva de registros por desastres ou encerramento de instituições – como ocorrido com a Universidade Gama Filho [Palma et al. 2019]. Essa falta de transparência e redundância cria brechas conhecidas e também utilizadas por agentes mal-intencionados, que criam falsas instituições especializadas na venda de certificados contrafeitos [Dias and Leal 2022].

É neste cenário que a Rede de Rastreabilidade de Dados da Educação Superior (RRDES), iniciativa do MEC em parceria com o Ministério da Economia e diversas universidades federais, disponibiliza um sistema que permite que discentes acompanhem suas trajetórias estudantis junto ao acesso a seus documentos acadêmicos pertinentes. Além disso, o projeto também tem o potencial de tornar-se uma plataforma conjunta para a emissão e registro destes certificados e até mesmo dados regulatórios das instituições de ensino superior [RNP 2023]. Em consonância a essa iniciativa, o presente estudo trata da implementação e validação de um protótipo de software de inteligência artificial, que combina diferentes técnicas de aprendizado de máquina, capaz de identificar imagens editadas de certificados antes de sua inserção neste ambiente.

2. Trabalhos Relacionados

A pesquisa sobre detecção de fraudes em documentos é ampla e o uso de inteligência artificial neste campo tem crescido significativamente, no entanto, a detecção de documentos acadêmicos falsificados permanece pouco explorada na literatura. Enquanto a detecção de fraudes foca em adulterações de arquivos originais, a de documentos falsificados busca identificar aqueles completamente forjados desde sua criação. Embora distintos, os métodos se sobrepõem, permitindo que este estudo aproveite referências de ambas as áreas.

No domínio, predominam estratégias de visão computacional, por exemplo: [Jaiswal et al. 2022] utilizam autoencoders convolucionais sobre imagens hiperspectrais para identificar incompatibilidades entre tintas; [Alameri et al. 2023] propõem abordagem não supervisionada usando correlações espectrais para gerar redes ponderadas e aplicar clustering; e [James et al. 2020] reformulam o problema como comparação de grafos via OCR para detectar manipulação de pixels. Alternativamente, outros métodos também são propostos, como o trabalho de [Boonkrong 2024], que utiliza funções hash para verificação preventiva de documentos previamente validados.

As abordagens mais robustas combinam múltiplas tecnologias. [Jain and Wigington 2019] demonstram eficácia da análise multimodal no dataset RVL-CDIP, combinando OCR, representações textuais (ULMFiT, FastText, n-grams) e visuais (VGG-16), alcançando alta acurácia. [Mohammed et al. 2024] aplicam K-means sobre features visuais de milhares de certificados para detecção de anomalias, identificando documentos distantes dos centroides com bons resultados.

3. Proposta

A solução proposta é um protótipo que, a partir da imagem de um documento acadêmico, decide se ele é suspeito de adulteração. Seu núcleo de detecção apoia-se no conceito da forense digital de que toda captura carrega propriedades que funcionam como uma assinatura. Tomando como exemplo o ruído residual, trata-se de um sinal espacialmente regular e uniforme que deriva das características da própria captura e digitalização, como imperfeições da lente, a incidência de luz sobre o sensor ou, no plano dos dados, os padrões deixados pela compressão da imagem. Quando um campo do documento é editado e reinserido, a região alterada rompe essa regularidade, pois passa a carregar o ruído de sua própria origem em vez daquele que revestia o documento autêntico, e é justamente essa anomalia estatística, localizada e destoante do entorno, que um modelo de aprendizado de máquina é capaz de aprender a detectar.

Esse raciocínio ancora-se em um modelo de ameaça no qual o fraudador não constrói um documento do zero, mas reaproveita um exemplar legítimo como base e nele substitui campos pontuais. Sob essa premissa, encadeiam-se dois sinais que partem de um mesmo documento e atuam em sequência. O primeiro é um gate de pertencimento, derivado do layout, que confronta o documento com as famílias de template conhecidas e o encaminha à revisão humana quando não se associa a nenhuma delas. O segundo, núcleo da detecção, é um detector forense supervisionado que varre o documento em nível de pixel e estima, região a região, a probabilidade de manipulação. As duas avaliações cobrem superfícies complementares, em que o detector flagra a edição feita sobre um template conhecido, enquanto o gate rejeita documentos de modelo desconhecido, legítimos ou não.

3.1. Pipeline de Treinamento

O fluxo de treinamento parte dos documentos convertidos em imagem e encadeia quatro etapas. A extração de layout e o agrupamento descobrem as famílias de template presentes no dataset, essas famílias orientam a geração das falsificações sintéticas que fornecem o material rotulado de treino, e sobre esse material retreina-se o detector forense. As subseções seguintes detalham cada etapa.

3.1.1. Topologia do Dataset

O dataset foi construído a partir de 2161 arquivos XML fornecidos pela UFSC, no esquema de diploma digital do MEC, dos quais 2066 traziam documentos embutidos. Deles selecionaram-se os de interesse deste trabalho, isto é, os históricos escolares e certificados de conclusão, codificados em base64 e recuperados como PDF. Cada PDF foi rasterizado, uma imagem por página, a 200 DPI em formato PNG sem perdas, resolução que preserva detalhes finos e evita artefatos de compressão que mais tarde poderiam ser confundidos com vestígios de manipulação. Converteram-se todas as páginas, e não apenas a de rosto, de modo que cada página é um item independente. Por fim, como um mesmo documento podia ter sido reenviado em arquivos distintos, o conjunto passou por deduplicação, que removeu 116 das 2105 imagens geradas e deixou os 1989 documentos sobre os quais operam as demais etapas. Esse conjunto é internamente heterogêneo, com documentos de instituições diversas distribuídos por algumas dezenas de famílias de template cujos números de exemplares variam drasticamente.

3.1.2. Extração de Layout

De cada documento extrai-se um descritor de layout, um vetor que resume sua estrutura visual e o torna comparável aos demais. O extrator emprega o DINOv2-large [Oquab et al. 2024] congelado como codificador, um vision transformer que reparte a imagem, redimensionada e preenchida a 1036×1036 pixels, em uma grade de 74×74 blocos e converte cada um em um vetor de 1024 dimensões. Escolheu-se o DINOv2 após avaliar outros codificadores, pois seu pré-treinamento auto-supervisionado sob objetivo discriminativo aproxima as representações de imagens semelhantes e afasta as de imagens distintas, o que se mostrou mais relevante para a tarefa do que a familiaridade com documentos oferecida por modelos treinados especificamente sobre eles. Sobre a grade de tokens aplica-se a agregação regional de características (R-MAC) de [Tolias et al. 2016], em três escalas de janelas que vão da página inteira a porções locais, tomando o maior valor de cada dimensão dentro de cada região, de modo que um logotipo ou selo que ative fortemente uma posição sobreviva à agregação. Os vetores regionais, normalizados, somam-se em um único descritor de 1024 dimensões, ao qual se aplica por fim o PCA-whitening, transformação aprendida uma única vez sobre todo o dataset que descorrelaciona as dimensões e equaliza suas variâncias, apontada como decisiva para o desempenho do R-MAC.

3.1.3. Agrupamento de Layout

O agrupamento inicia truncando os descritores a suas 128 primeiras dimensões, as de maior variância segundo o PCA, o que descarta os eixos de baixa variância que apenas injetariam ruído nas comparações. Sobre esses vetores aplica-se uma clusterização aglomerativa com linkage completo e distância de cosseno, com limiar de corte de 0,55, o que mantém todo par de um mesmo grupo abaixo dessa distância e forma grupos enxutos e coesos, cada um reunindo instâncias muito próximas de um template, enquanto os documentos sem vizinhos próximos o suficiente ficam como ruído. Como um mesmo modelo aparece sob diferentes gerações de impressão e digitalização que o codificador

trata como distintas, um segundo estágio reagrupa os próprios clusters para reconstituir as famílias, representando cada grupo por seu centroide e seu medoide e unindo os que estão próximos, medoides a até 0,50 ou centroides a até 0,45 de distância, dessa vez admitindo o encadeamento transitivo que reconecta capturas distantes de um mesmo modelo. Esses parâmetros não foram arbitrados, mas decorrem de uma busca experimental que variou o critério de linkage, o limiar de corte e o comprimento do truncamento, avaliando a cada ajuste a coesão dos grupos e a coerência visual dos agrupamentos. Ao final, os clusters de segunda ordem orientam a geração de falsificações, enquanto os documentos deixados como ruído servem à calibragem do gate de inferência.

3.1.4. Geração de Falsificações

O detector forense aprende de exemplos rotulados, isto é, documentos falsificados acompanhados da indicação exata dos pixels alterados, material que, na ausência de fraudes reais no dataset, é gerado sinteticamente. Primeiro localizam-se os campos de texto de cada documento por OCR, com a biblioteca EasyOCR e um limiar de confiança permissivo de 0,3, que fornece as caixas sobre as quais uma edição pode ser aplicada. A partir disso definem-se três classes de falsificação. A classe A substitui um campo por outro compatível em posição e tamanho retirado de um documento da mesma família, produzindo uma adulteração coerente e difícil de notar. A classe B faz a mesma troca, porém com um campo vindo de outra família, o que provoca uma incompatibilidade visível de tipografia e reforça o aprendizado de quebras de contexto. A classe C edita o próprio documento por substituição de conteúdo textual, cópia de um trecho para outra posição, apagamento por preenchimento com o fundo ou cobertura por um retalho neutro, reproduzindo o ataque tal como se presume ocorrer na prática. Cada documento real dá origem a cerca de cinco falsificações, cada uma composta pela imagem adulterada, sua máscara de adulteração e os metadados de proveniência. O conjunto resultante é dividido em treino, validação e teste na proporção aproximada de 80%, 10% e 10%, mantendo cada família de template inteira em um único conjunto e separando por documento de origem, o que impede qualquer vazamento entre treino e teste.

3.1.5. Treinamento do Detector

O núcleo de detecção é o ASC-Former, arquitetura forense voltada à detecção de adulteração de texto em documentos [Luo et al. 2024]. Em vez de treiná-lo do zero, parte-se dos pesos públicos disponibilizados por seus autores, já treinados sobre o dataset RTM, com mais de quatorze mil imagens de documentos autênticos e adulterados, e como o RTM trata exatamente da mesma tarefa, esses pesos transferem bem, restando apenas um fine-tuning que retreina todas as camadas para adaptá-las aos documentos acadêmicos e às falsificações produzidas. O treino cobre as três classes de falsificação somadas aos documentos reais autênticos, indispensáveis para ensinar o modelo a não disparar sobre páginas legítimas, na proporção de um autêntico para cinco adulterados. Como uma página a 200 DPI é custosa demais para ser processada de uma vez, cada passo opera sobre um recorte de 512×512 pixels, metade deles deliberadamente centrada sobre a região adulterada para que o modelo a veja com frequência suficiente. A aumento de dados restringe-se ao espelhamento horizontal, pois perturbações mais agressivas destrui-

riam justamente os vestígios que se deseja detectar. Quanto aos demais hiperparâmetros, reproduz-se o método do artigo original [Luo et al. 2024], com a mesma função de perda, otimizador e cronograma de 80 mil iterações, sob a premissa de que empregar a arquitetura nas condições para as quais foi validada reduz o risco de um ajuste malsucedido. Para converter o mapa de probabilidade produzido pelo detector em uma decisão por documento, toma-se a média dos pixels de maior resposta, agregação conhecida como mean top- k , elevada quando há uma região concentrada de forte resposta e baixa quando nenhuma se destaca, e o limiar que separa suspeitos de legítimos é fixado uma única vez sobre o conjunto de validação, no ponto que maximiza a estatística J de Youden.

3.2. Pipeline de Inferência

Na inferência, o documento submetido passa primeiro pela extração de layout, cujo descritor, após o mesmo branqueamento e truncamento ajustados no treino, é atribuído por proximidade à família de template mais próxima, medida pela distância de cosseno a seu mediano. Quando fica distante de todas, dispara-se o gate de pertencimento e o documento é tido como de modelo desconhecido, encaminhado à revisão de um especialista, pois diante de um modelo nunca visto não há garantia sobre quais adulterações seriam plausíveis nem base para calibrá-las. Reconhecida a família, o documento segue ao ramo forense, em que o detector varre a imagem inteira e devolve o mapa de adulteração em nível de pixel, condensado no score e comparado ao limiar fixado no treino, classificando-se o documento como suspeito quando o score o ultrapassa e como normal caso contrário. O resultado é sempre acompanhado do mapa de adulteração, que evidencia as regiões responsáveis pela decisão e confere explicabilidade ao sistema. Esse encadeamento condiciona a detecção forense ao reconhecimento prévio do layout, o que restringe a avaliação aos templates já conhecidos, mas preserva uma propriedade desejável, pois vincula a solução ao modelo de ameaça, e não a tipos específicos de documento, de modo que novas famílias possam ser incorporadas por retreinamento à medida que documentos desconhecidos se acumulam e se confirmam.

4. Resultados

O agrupamento de layout sobre os 1989 documentos formou 473 clusters no primeiro estágio e deixou 38,1% dos documentos como ruído, sem vizinhos próximos o suficiente para agrupar. O segundo estágio reuniu clusters dispersos em 83 famílias de template, confirmando o comportamento esperado de que um mesmo modelo se fragmenta em vários grupos por variações de impressão e digitalização. A validação foi também qualitativa, adotando-se como critério que ao menos os vinte maiores clusters fossem visualmente homogêneos quanto ao modelo, condição que se verificou.

A geração de falsificações produziu 16.093 amostras a partir desses documentos, distribuídas entre as três classes, cada uma composta pela imagem adulterada, sua máscara e os metadados da edição. As manipulações têm caráter pontual, cobrindo em média pouco mais de 1% da página nas trocas de campo e apenas 0,2% nas edições internas da classe C, o que torna a detecção tanto mais difícil quanto menor a adulteração.

O desempenho do detector forense foi medido sobre o conjunto de teste, por classe de falsificação, e é resumido na Tabela 1. A métrica central é o F1 da classe C, que reproduz o modelo de ameaça mais realista e alcança 0,850, ao passo que as classes A

e B, de troca de campo e mais fáceis de detectar, chegam a 0,958 e 0,973. A classe C combina precisão muito alta, de 0,978, com recall mais modesto, de 0,751, o que revela um detector conservador, que raramente acusa em falso mas ainda deixa escapar cerca de um quarto das falsificações mais sutis. A separação entre falsificações e autênticos é forte nas três classes, com AUROC de 0,900 na classe C a 0,993 na classe B, a taxa de falsos positivos sobre documentos legítimos ficou em 8,0%, e, quando a falsificação é detectada, sua localização em nível de pixel permanece precisa.

Tabela 1. Desempenho do detector forense a nível de documento, por classe de falsificação

Classe	F1	Precisão	Recall	AUROC
A	0,958	0,957	0,959	0,985
B	0,973	0,974	0,973	0,993
C	0,850	0,978	0,751	0,900

5. Considerações Finais

Este trabalho apresentou um protótipo capaz de classificar documentos acadêmicos quanto à suspeita de fraude, concebido como ferramenta de apoio à decisão de um especialista, e não como veredito final. O pipeline que encadeia o agrupamento de templates, a geração de falsificações sintéticas e o detector forense mostrou-se eficaz, alcançando F1 de 0,850 na classe de falsificações mais realista e valores superiores nas demais, ainda que se identifique como principal limitação a menor sensibilidade a adulterações muito pequenas. Como o detector especializa, para o domínio acadêmico, a arquitetura ASC-Former pré-treinada sobre o dataset RTM, o F1 de 0,807 que o artigo original reporta sobre esse conjunto [Luo et al. 2024] serve de referência, e o resultado obtido mostra-se compatível com tal patamar, indício de que a especialização transferiu bem para o caso de uso, sem que isso represente, dadas as diferenças de dataset e de classes de falsificação, uma superação do modelo original.

Por ter sido concebida em torno do modelo de ameaça, e não de um tipo específico de documento, a solução pode operar sobre quaisquer famílias que nele se enquadrem e ser periodicamente retreinada para incorporar layouts que ainda não reconhece, à medida que se acumulam.

Referências

Alameri, M., Mahmood, B., Ciylan, B., and Amged, A. (2023). Unsupervised forgery detection of documents: A network-inspired approach. *Electronics*, 12.

Boonkrong, S. (2024). Design of an academic document forgery detection system. *International Journal of Information Technology*, pages 1–13.

Dias, P. and Leal, A. (2022). Sites vendem diploma de curso superior para quem sequer pisou em sala de aula: 'documentação 100% original, emitida de dentro da universidade', diz atendente. *O Globo*. Disponível em: <https://oglobo.globo.com/brasil/noticia/2022/11/sites-vendem-diploma-de-curso-superior-para-pessoas-que-nao-concluíram-ou-sequer-pisaram-em-uma-universidade.ghml>. Acesso em: 05 abr. 2025.

- INEP (2024). Censo da educação superior 2023: notas estatísticas.
- Jain, R. and Wigington, C. (2019). Multimodal document image classification. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 71–77.
- Jaiswal, G., Sharma, A., and Yadav, S. (2022). Deep feature extraction for document forgery detection with convolutional autoencoders. *Computers & Electrical Engineering*, 99:107770.
- James, H., Gupta, O., and Raviv, D. (2020). Ocr graph features for manipulation detection in documents. *arXiv preprint arXiv:2009.05158*.
- Luo, D., Liu, Y., Yang, R., Liu, X., Zeng, J., Zhou, Y., and Bai, X. (2024). Toward real text manipulation detection: New dataset and new solution.
- Mohammed, S., Nwobodo, L., and Ekene, N. (2024). Certificate fraud verification model using clustered-based classification approach. *Explorematics Journal of Innovative Engineering and Technology*, 5(1):60–72.
- Oquab, M., Darcet, T., Moutakanni, T., et al. (2024). Dinov2: Learning robust visual features without supervision.
- Palma, L. M., Vigil, M. A. G., Pereira, F. L., and Martina, J. E. (2019). Blockchain and smart contracts for higher education registry in brazil. *International Journal of Network Management*, 29.
- RNP (2023). Blockchain da jornada acadêmica. Youtube. Disponível em: <https://www.youtube.com/watch?v=xqezMbjCeTM>. Acesso em: 13 mai. 2025.
- Tolias, G., Sicre, R., and Jégou, H. (2016). Particular object retrieval with integral max-pooling of cnn activations.