Arthur Dimitri Brito Oliveira

Relatório de Estágio Supervisionado LIHM - Laboratório de Interface Homem-Máquina

Campina Grande, Brasil

18 de maio de 2021

Arthur Dimitri Brito Oliveira

Relatório de Estágio Supervisionado LIHM - Laboratório de Interface Homem-Máquina

Relatório de estágio supervisionado submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Universidade Federal de Campina Grande - UFCG

Centro de Engenharia Elétrica e Informática - CEEI

Departamento de Engenharia Elétrica - DEE

Orientador: Danilo Freire de Souza Santos, D. Sc.

Campina Grande, Brasil

18 de maio de 2021

Arthur Dimitri Brito Oliveira

Relatório de Estágio Supervisionado LIHM - Laboratório de Interface Homem-Máquina

Relatório de estágio supervisionado submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Trabalho aprovado em: / /

Danilo Freire de Souza Santos, D. Sc.
Orientador

Saulo Oliveira Dornellas Luiz, D. Sc.
Convidado

Campina Grande, Brasil
18 de maio de 2021

Agradecimentos

Gratidão a Deus pelo fôlego de vida diário que me permitiu chegar até aqui.

Agradeço aos meus pais, Agostinho e Luciana, que nunca mediram esforços e incentivos durante toda a minha formação. Espero continuar retribuindo todo o suporte fornecido ao longo de todos esses anos.

A Débora, minha namorada, minha gratidão pela companhia nas inúmeras idas à UFCG e pelas palavras de ânimo nos dias em que tudo parecia ir de mal a pior.

Ao professor Danilo, agradeço pela contínua disponibilidade e prontidão nos momentos que precisei. Ao professor Gutemberg Júnior, gratidão pela paciência e dedicação incansáveis na resolução das burocracias envolvidas na formalização deste estágio.

Agradeço ao VIRTUS como um todo, especialmente a Wislayne, Bruna e a meu ex-companheiro de equipe André, pelo suporte diário ao longo deste ano atípico, pela oportunidade de vivência em um ambiente profissional e pela imensurável contribuição para minha formação.

Lista de ilustrações

Lista de abreviaturas e siglas

API	Application Programming Interface
CEEI	Centro de Engenharia Elétrica e Informática
CNN	Convolutional Neural Network
CPU	Central Process Unit
GPU	Graphics Processing Unit
LIHM	Laboratório de Interface Homem-Máquina
NLP	Natural Language Processing
OCR	Optical Character Recognition
SOTA	State of The Art
UFCG	Universidade Federal de Campina Grande
TFIDF	Term Frequency Inverse Document Frequency
VL	Visual Linguistic

Sumário

1 Introdução

Neste relatório são apresentadas as atividades desenvolvidas pelo aluno Arthur Dimitri Brito Oliveira no Estágio Supervisionado. Estas atividades foram realizadas no contexto de um projeto de PDI mediante uma parceria entre o Laboratório de Interface Homem-Máquina (LIHM/UAEE) e o VIRTUS UFCG - Núcleo de Pesquisa, Desenvolvimento e Inovação em Tecnologia da Informação, Comunicação e Automação, ambos da Universidade Federal de Campina Grande (UFCG) [1]. O VIRTUS é um órgão suplementar da Universidade Federal de Campina Grande, especificamente do Centro de Engenharia Elétrica e Informática (CEEI). Durante o período que compreende 03/02/2021 a 09/04/2021, as atividades ocorreram no âmbito de um projeto de Pesquisa e Desenvolvimento. O estágio foi realizado sob orientação do professor Danilo Freire de Souza Santos e supervisionado pelo professor Jaidilson Jó da Silva.

A disciplina de estágio (supervisionado ou integrado) é uma disciplina compulsória na grade do curso de Engenharia Elétrica na Universidade Federal de Campina Grande. O intuito desta oportunidade é disponibilizar ao aluno um contexto de aplicação dos conhecimentos teóricos adquiridos durante a realização do curso, estreitar o contato com profissionais da área de atuação à qual a atividade se propõe e a aquisição de conhecimentos comumente desejáveis no mercado de trabalho.

As atividades contemplam um total de 180 horas, divididas em carga horária de 20 horas semanais, onde o estagiário realizou atividades presenciais nas instalações do Laboratório de Interface Homem-Máquina e parte de forma remota, devido às restrições estaduais impostas pela pandemia global de Covid-19.

A atuação do estagiário durante o período em questão incluiram a exploração de modelos de aprendizagem de máquina com foco em visão computacional, desenvolvimento de casos de uso para validação de soluções de aprendizado de máquina e a imersão em processos de gestão e desenvolvimento ágil de sistemas.

[1] Endereço do VIRTUS: R. Aprígio Veloso, 1500 - Bodocongó, Campina Grande - PB

2 O laboratório - LIHM

O Laboratório de Interfaces Homem-Máquina (LIHM) compreende um ambiente de pesquisas e desenvolvimento no qual são realizadas atividades discentes nos níveis de graduação e pós-graduação. No laboratório realizam-se testes e experimentos relacionados às disciplinas e pesquisas acadêmicas. As pesquisas são voltadas para o desenvolvimento e avaliação de sistemas e produtos de *hardware* e *software*, com foco na interface do usuário com sistemas e produtos utilizados em ambientes de automação industrial, visando a redução do erro humano na operação destes sistemas.

O LIHM desenvolve parcerias com outros laboratórios, como o VIRTUS e o Laboratório Embedded. Existem projetos de PDI (Pesquisa, Desenvolvimento e Inovação) em diversas áreas, como desenvolvimento de *Software*, Inteligência Artificial e *Hardware*. No contexto de alunos graduandos, a parceria abrange, também, projetos de capacitação, onde os estudantes têm a oportunidade de vivenciar um ambiente real de projeto.

3 Metodologia de Desenvolvimento

Esta seção descreve as etapas e papéis envolvidos na metodologia utilizada para a realização das atividades de estágio.

3.1 Desenvolvimento ágil utilizando Scrum

No ambiente de desenvolvimento do projeto, as atividades foram desenvolvidas utilizando a metodologia Scrum. Este termo descreve um arcabouço conceitual para gerenciamento ágil de projetos no geral (SCHWABER; SUTHERLAND, 2020). Sua base é a composição de pequenas equipes com diferentes frentes, que trabalham de forma cooperativa com comunicação constante.

Diante de inúmeras demandas partindo do cliente, é possível orientar o trabalho das equipes para focar em resultados que agreguem valor de forma contínua por meio de entregas parciais. Além disso, neste processo, as expectativas são niveladas entre os participantes do projeto e os requisitos de entrega desejados pelo cliente.

O Scrum é uma metodologia que define responsabilidades para os integrantes, de tal forma que cada um conhece a sua devida função. E, a partir desta premissa, um conceito muito importante surge: o auto-gerenciamento. Quando cada componente sabe exatamente quais são as suas obrigações, a equipe se auto-gerencia e isto contribui para um andamento mais fluido dos projetos.

A *Sprint* é um contêiner de todos os eventos no universo Scrum. Cada evento no Scrum é uma oportunidade de inspecionar e adaptar os artefatos gerados. Estes eventos são concebidos especificamente para permitir a transparência necessária no processo de desenvolvimento. Os eventos criam uma regularidade no processo e minimizam a necessidade de reuniões que não estejam pré-definidas no Scrum.

A unidade fundamental do Scrum é o *Scrum Team*. Ele é composto por *Developers*, um *Scrum Master* e um *Product Owner*. Dentro de um time, não há sub-times ou hierarquias. Trata-se de uma unidade coesa de profissionais focados no objetivo atual do produto.

Os *Developers* são responsáveis pela criação de qualquer aspecto incremental da solução ao longo da *Sprint*. No geral, as atividades dessa parcela do time estão associadas à criação de um plano para atender as demandas e à adaptação diária deste em direção ao objetivo da *Sprint*. O *Scrum Master* é responsável por garantir a execução do método ágil seguindo as diretrizes Scrum. Ele lidera o time e o conduz, garantindo que haja auto-gerenciamento, entregas parciais com incremento de valor dentro dos prazos estimados e

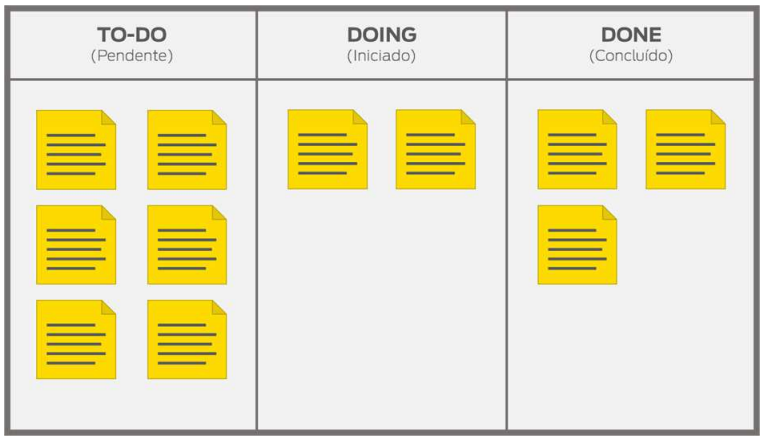
auxilia na resolução de possíveis impedimentos. Por fim, o *Product Owner* é responsável por maximizar o valor do produto resultante do trabalho do *Scrum Team*. Define e explicita o objetivo do produto, comunica os requisitos relacionados às entregas parciais e garante que as requisições são claras, visíveis e compreendidas por todos.

No que diz respeito aos eventos, a *Sprint Planning* inicia a *Sprint* ao definir o trabalho a ser realizado durante a mesma. O planejamento resultante é criado de forma colaborativa pelo *Scrum Team* e deve ser voltado à estimação do esforço empregado na resolução de tarefas que apontem para o objetivo do produto. O outro tipo de reunião é a *Daily Scrum*. Nela inspeciona-se o progresso em direção ao objetivo da *Sprint*. É uma reunião rápida, que dura em média 15 minutos, e envolve o time de *Developers*.

Ao final, há a *Sprint Review* e a *Sprint Retrospective*. Na *Sprint Review* o time apresenta os resultados do seu trabalho para as partes interessadas e há uma discussão quanto ao progresso em direção aos objetivos do produto. Já a *Sprint Retrospective* é utilizada para planejar maneiras de incrementar a qualidade e a efetividade do time. Nela, o *Scrum Team* inspeciona como a sprint ocorreu no que tange os indivíduos, interações, processos e ferramentas.

Basicamente, existem três estados para descrever as tarefas propostas. *To Do* no caso das tarefas ainda não iniciadas, *Doing* para as tarefas em andamento e *Done* para as tarefas já realizadas. Estes compõem o quadro Scrum, que pode ser ilustrado na Figura 1. O quadro condensa as informações essenciais do projeto.

Figura 1 – Quadro Scrum.



Fonte: o próprio autor.

3.2 O papel do estagiário

O estagiário estava enquadrado na equipe de *Developers*. Especificamente, interagiu com equipes de aprendizagem de máquina e visão computacional, absorvendo muito da experiência prática do time. Imerso no processo de desenvolvimento ágil, foi respon-

sável pela investigação de arquiteturas e tecnologias para implementação de sistemas de aprendizagem de máquina com foco em visão computacional. Além disso, participou na experimentação e validação de subsistemas de visão computacional.

3.3 Execução da metodologia

A vivência no ambiente de projeto permitiu experienciar múltiplas etapas presentes na metodologia Scrum. Inicialmente, as demandas por parte do *Product Owner* eram definidas e havia uma reunião de planejamento, coordenada pelo *Scrum Master*, com a participação dos *Developers*. Nesta reunião, as demandas eram divididas em tarefas menores e havia uma votação para definir o esforço estimado para cada uma delas.

Durante a *Sprint*, houve a participação contínua nas reuniões diárias, *Daily Scrum*, conduzidas pelo *Scrum Master*. Nessas reuniões, reportava-se de forma objetiva o que havia sido feito no dia anterior e qual era o planejamento para o dia atual. Além disso, impedimentos que comprometeram a resolução das tarefas também eram relatados.

Ao final de um período estimado em quinze dias, havia a *Sprint Review*. Nela, os resultados para as demandas propostas eram apresentados diretamente ao *Product Owner*. Em sequência, normalmente realizava-se a *Sprint Retrospective* do período, elencando pontos positivos, negativos e de ação. Este momento foi muito importante na vivência da experiência profissional, visto que era possível acompanhar de forma contínua se os pontos de correção propostos eram efetivamente corrigidos pela equipe.

O registro contínuo de atividades foi feito utilizando a ferramenta de gestão de projetos Turmalina. Esta ferramenta possibilita aos *Scrum Masters* uma visão geral do andamento do projeto, compreendendo as etapas *To Do*, *Doing* e *Done*, além dos impedimentos associados.

4 Fundamentação Teórica

Nesta seção são descritos os conceitos fundamentais relacionados à aprendizagem multimodal. Tais conceitos permeiam a realização das atividades propostas no estágio.

4.1 Aprendizagem de máquina multimodal

A palavra modalidade refere-se à maneira como algo acontece ou é vivenciado. Um problema de pesquisa é caracterizado como multimodal quando inclui várias modalidades (GALLO I.; JANJUA, 2019). A aprendizagem de máquina multimodal visa, então, construir modelos capazes de processar e relacionar informações de múltiplas modalidades, como textos, sons, vídeos e imagens.

Estratégias multimodais são especialmente úteis em cenários onde há ambiguidade no domínio do problema de classificação, sendo capazes de prover melhorias no desempenho do sistema, já que a abordagem multimodal pode capturar relações entre representações de diferentes domínios (BALTRUŠAITIS; AHUJA; MORENCY, 2017).

Embora esse campo possua um enorme potencial de exploração, existem diversos desafios a serem enfrentados, que podem ser divididos nos seguintes tópicos:

- **Representação**: O primeiro desafio fundamental é aprender como representar e sumarizar dados multimodais, de modo a explorar sua complementaridade.
- **Tradução**: O segundo desafio é traduzir, ou mapear, uma modalidade para outra. A relação entre modalidades é geralmente vaga, aberta e subjetiva.
- **Alinhamento**: Identificar as relações diretas entre elementos e subelementos de diferentes modalidades.
- **Fusão**: O quarto desafio é a fusão de informações entre modalidades para a realização de predições. Deve-se avaliar qual tipo de fusão ocasionará em um maior benefício para o sistema de predição.
- **Co-aprendizagem**: O ultimo desafio é a transferência de conhecimento entre as modalidades, suas representações e seus modelos preditivos.

Representar dados, de tal forma que modelos computacionais possam processá-los, sempre foi um desafio. O trabalho proposto por (BAHDANAU; CHO; BENGIO, 2014) utiliza os termos *feature* e *representation* de forma intercambiável, referindo-se a um vetor ou tensor que representa uma entidade (imagem, texto ou sinal). A habilidade

de representar os dados de modo significativo é crucial para problemas multimodais. Há, também, algumas propriedades necessárias para uma boa representação, entre elas a suavidade, as coerências espacial e temporal e a esparsidade.

Os avanços no contexto de mídias unimodais têm sido bastante significativos, como na NLP (*Natural Language Processing)*, da simples ocorrência de termos TF-IDF (*Term Frequency–Inverse Document Frequency*) a representações que levam em conta o contexto semântico com mapeamento no espaço vetorial utilizando *Word Embeddings*. Em abordagens multimodais do passado, as representações envolviam simples concatenações unimodais, mas outros tipos de soluções têm sido propostas na última década.

4.1.1 Técnicas de fusão

A fusão multimodal é um dos tópicos originais da aprendizagem de máquina multimodal. Se resume a integrar informações de múltiplas modalidades com o objetivo de classificação ou regressão.

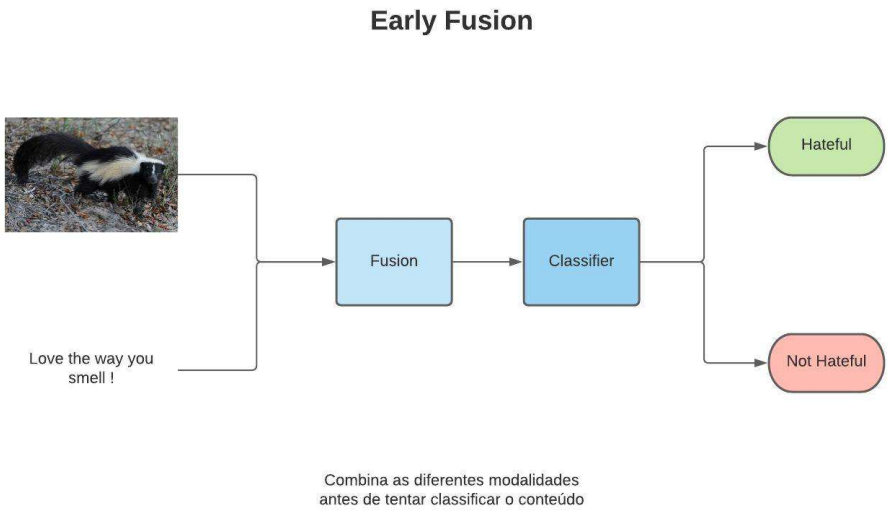
O interesse se explica pelo fato de que ter acesso a múltiplas modalidades que observam o mesmo fenômeno pode contribuir para predições mais robustas (BALTRUŠAITIS; AHUJA; MORENCY, 2017). Ademais, a multimodalidade pode prover informações complementares que não seriam visíveis nas modalidades de forma isolada, permitindo a operação na ausência de uma delas.

Diz-se que há um mecanismo de fusão quando a integração multimodal é feita nos estágios iniciais ou finais de predição. Atualmente, a linha existente entre representações multimodais e fusão se tornou ainda mais tênue, especialmente quando trata-se de modelos que envolvem aprendizagem profunda, onde o aprendizado por representação interage com os objetivos de classificação ou regressão.

No caso das abordagens *model-based*, onde a fusão tem papel fundamental no processo de modelagem, existem duas subcategorias quanto à união das modalidades: fusão antecipada e fusão tardia. Na fusão antecipada, as modalidades são concatenadas antes da utilização do classificador, seja na forma de concatenação vetorial, ou por meio de operações mais complexas, como o produto Hadamard (DUKE; TAYLOR, 2018).

A fusão antecipada pode ser vista como uma tentativa mais direta dos pesquisadores para realizar uma representação multimodal, já que ela é capaz de explorar as interações e correlações entre *features* de baixo nível de cada modalidade. Além disso, esta técnica só requer o treinamento de um único modelo, tornando o *pipeline* mais simples quando comparado às fusões tardia e híbrida. A ilustração do processo de fusão antecipada pode ser visualizada na Figura 2.

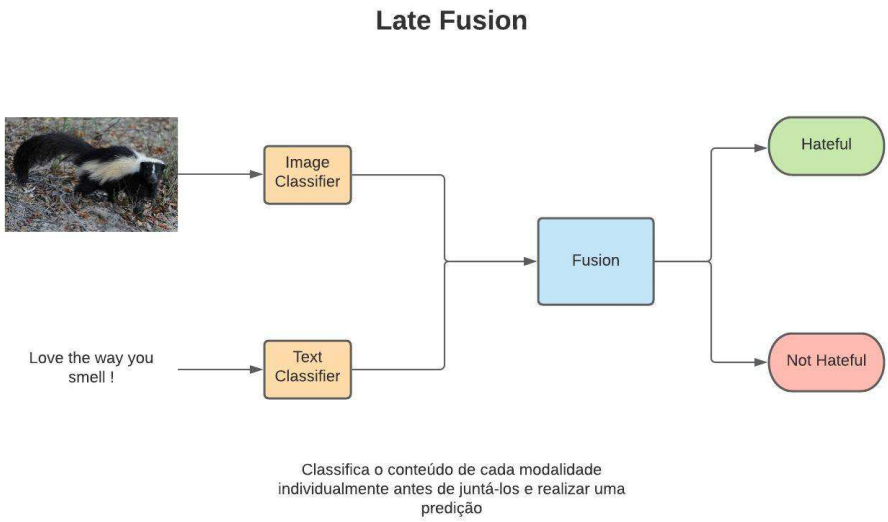
Figura 2 – Diagrama da aplicação da fusão antecipada.



Fonte: o próprio autor.

A fusão tardia, por sua vez, utiliza utiliza valores de decisão unimodais como parte de um mecanismo de decisão. Posteriormente, utilizando algum método de concatenação, aplica-se a fusão tardia antes do classificador final. Essa abordagem torna mais fácil realizar predições quando uma das modalidades está ausente. No entanto, a fusão tardia ignora completamente as interações a baixo nível entre as modalidades. A ilustração do processo pode ser observada na Figura 3

Figura 3 – Diagrama de fusão tardia



Fonte: o próprio autor.

5 Atividades Desenvolvidas

Nesta seção detalha-se as ferramentas utilizadas e as atividades desenvolvidas durante o período de estágio. Devido ao caráter de confidencialidade presente em cláusulas contratuais no acordo de parceria do projeto, as atividades listadas e objetivos finais são descritos em alto nível neste relatório.

5.1 Objetivo

O principal objetivo do estágio foi a investigação, experimentação e documentação de abordagens para classificação de conteúdos multimodais utilizando aprendizagem profunda. Neste processo, aplicou-se métodos ágeis de desenvolvimento.

5.2 *Frameworks* utilizados

Frameworks podem ser definidos como um conjuntos de bibliotecas que contém diversas funções prontas a serem utilizadas em um ambiente de desenvolvimento. Devido à sua natureza genérica, sua utilização pode se dar em diferentes contextos de programação. Ao longo do período de estágio, utilizou-se alguns *frameworks* descritos a seguir.

5.2.1 Keras e TensorFlow

TensorFlow (TENSORFLOW, 2020) é uma biblioteca de código aberto para aprendizado de máquina aplicável a uma ampla variedade de tarefas. É um sistema para criação e treinamento de redes neurais visando a detecção de padrões e correlações. Keras (CHOLLET, 2020) é uma API (*Application Programming Interface)* de alto nível construído sobre o TensorFlow, que traz uma experiência mais amigável para o usuário.

O Keras traz um conceito de alta modularidade, sendo possível desenvolver o protótipo de um modelo de aprendizagem de máquina de forma rápida. No caso de experimentos onde utiliza-se arquiteturas já conhecidas de redes neurais, é a API mais indicada. O TensorFlow é altamente recomendado quando deseja-se um monitoramento contínuo do comportamento da rede. Além disso, fornece operações mais complexas do que o Keras e possui um *debugger* especializado.

5.2.2 Pytorch

Pytorch (PYTORCH, 2020) é um *framework* de código-aberto de aprendizagem de máquina que acelera o processo entre a prototipagem e o *deploy* dos modelos no ambiente

de produção. O *backend* permite treinamento e otimização de modelos de forma escalável e distribuída. É composto por um amplo conjunto de bibliotecas e ferramentas que permite ao Pytorch o suporte no desenvolvimento de projetos envolvendo visão computacional, NLP e outros, inclusive em plataformas de computação na nuvem.

5.2.3 SpaCy

SpaCy [1] é uma ferramenta código aberto para processamento avançado de linguagem natural, escrita na linguagem de programação Python. Contém *pipelines* baseados em *Transformers*, o que permite elevar o desempenho na resolução de tarefas de NLP ao patamar de estado da arte.

5.3 Descrição das atividades desenvolvidas

O objetivo desta etapa do estágio foi apresentar uma revisão bibliográfica e experimentação de abordagens acadêmicas para classificação de imagens e textos utilizando o Google Colab [2]. Nesta seção encontram-se as descrições das atividades realizadas. Realizou-se inicialmente a investigação de abordagens envolvendo fusão multimodal, e em seguida a investigação de abordagens multimodais para lidar com conteúdo impróprio. Posteriormente, realizou-se a investigação de ferramentas de código-aberto para reconhecimento ótico de caracteres. Por fim, realizou-se a atividade de investigação de abordagens utilizando *Transformers* VL (*Visual Linguistic).*

5.3.1 Investigação de abordagens envolvendo fusão multimodal

Diante do caráter experimental e investigativo desta etapa, buscou-se soluções acadêmicas para problemas multimodais. No trabalho proposto por (GALLO I.; JANJUA, 2019), propõe-se a transformação da descrição textual da imagem em uma representação visual enriquecida contendo *features* textuais e visuais. Utiliza-se uma CNN (*Convolutional Neural Network*), uma rede neural convolucional, que geralmente é utilizada somente para classificação de conteúdos unimodais.

A principal contribuição deste trabalho foi propor uma maneira alternativa de representar e combinar textos e imagens ao criar uma única imagem enriquecida, que pode ser treinada por uma CNN tradicional. No que tange a pesquisa, foi uma etapa importante para elucidar como seria possível utilizar esta abordagem para classificação de imagens com texto.

[1] spaCy 101: Everything you need to know. Acessado em: 07 de Maio de 2021. Disponível em: https://spacy.io/usage/spacy-101

[2] O que é o Google Colaboratory? | Google. Acessado em: 08 de Maio de 2021. Disponível em: https://colab.research.google.com/notebooks/intro.ipynb?hl=pt$_BR$

Na abordagem proposta, a descrição em texto é representada por meio de *word embeddings* e posteriormente codificada em uma matriz de *pixels* RGB. Posteriormente, essa matriz de pixels é superposta à imagem e o conteúdo é passado para um classificador unimodal. O código apresentado no repositório [3], no entanto, não contém uma documentação extensa. Desse modo, a extração das imagens e textos do *dataset* original e do *dataset* modificado precisou ser feita a partir da análise dos *scripts*. Os caminhos necessários para os módulos não eram enxergados pelos *scripts*. A formatação dos conjuntos de treino e validação não era clara, não sendo possível replicar os resultados e compreender a estrutura de arquivos, o preprocessamento e o *pipeline* utilizado. Não foi possível realizar predições utilizando exemplos inéditos devido à impossibilidade de acesso direto aos modelos pré-treinados com os conjuntos de dados mencionados.

A Tabela 1 apresenta resultados para conteúdos textuais, visuais e mistos. Nos conjuntos dados mistos, *Ferramenta* e *Food-101*, há imagens e suas descrições em texto. A primeira coluna da tabela apresenta o resultado utilizando apenas as *features* textuais. Em seguida, há o resultado em termos da acurácia para duas redes diferentes, utilizando apenas as imagens. Por fim, o resultado utilizando as imagens enriquecidas com *features* textuais. Os valores de acurácia reforçam a efetividade em utilizar-se técnicas multimodais para predição em relação a técnicas unimodais.

Tabela 1 – Resultados, em termos da acurácia, obtidos após replicação do código.

Dataset	**Texto**	**Imagem**		**Fusão**	
		AlexNet	GoogleNet	AlexNet	GoogleNet
Ferramenta	92.09	92.36	92.47	95.15	95.45
Food-101	79.78	42.01	55.65	82.90	83.37

Continuando a revisão bibliográfica, o estagiário investigou a abordagem proposta por (SETH; BISWAS, 2017). O comparativo é feito entre duas abordagens de fusão tardia: uma utilizando a fusão das *features* textuais após os classificadores unimodais, e outra realizando a fusão a partir das probabilidades dos classificadores unimodais. A esta última denominou-se Regra de Aprendizagem.

Utilizou-se um conjunto de dados contendo 1521 imagens de e-mails maliciosos e 1500 de e-mails comuns. Para o texto, utilizou-se o conjunto de dados *Enron Spam Dataset*, que contém 16537 mensagens comuns e 17108 spams. Os valores em termos da acurácia e do F1-score estão descritos na Tabela 2. O F1-score leva em consideração, de forma ponderada, a precisão e a revocação do modelo, métricas comuns no ramo da aprendizagem de máquina. Como pode-se observar, as estratégias multimodais têm desempenho superior quando comparadas às convencionais, especialmente no caso da

[3] Multimodal Classification. Disponível em: https://github.com/artelab/Multi-modal-classification. Acesso em: 08 de Maio de 2021.

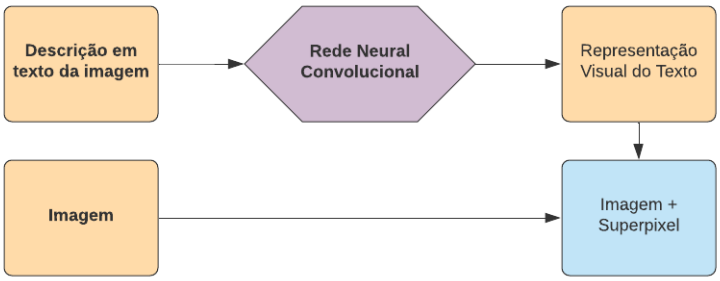
técnica que utiliza a Regra de Aprendizagem. Especialmente a abordagem utilizando a regra de aprendizagem (fusão tardia), que obteve um F1-score superior às outras soluções.

Tabela 2 – Resultados obtidos para as múltiplas abordagens.

	Acurácia	**F1-Score**
CNN-Imagem	85.89%	0.87
CNN-Texto	97.54%	0.97
Multimodal (Fusão)	96.87%	0.95
Multimodal (Regra de Aprendizagem)	98.11%	0.98

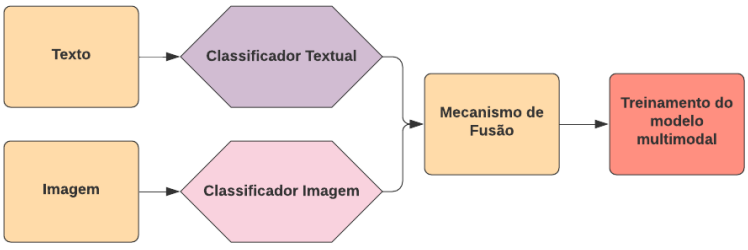
Após essa revisão bibliográfica, o estagiário pôde reportar em uma *Sprint Review* alguns métodos genéricos plausíveis para a resolução de problemas multimodais. A primeira abordagem proposta envolveria uma fusão antecipada com representação textual na forma de *superpixel* RGB e uma classificação posterior utilizando uma rede CNN, conforme ilustrado na Figura 4. A segunda abordagem utilizaria dois classificadores unimodais, um para imagem e outro para texto, um mecanismo de fusão tardia e, por fim, a classificação utilizando uma rede CNN. Esta última abordagem pode ser ilustrada na Figura 5.

Figura 4 – Abordagem proposta utilizando fusão antecipada.



Fonte: o próprio autor.

Figura 5 – Abordagem proposta utilizando fusão tardia.

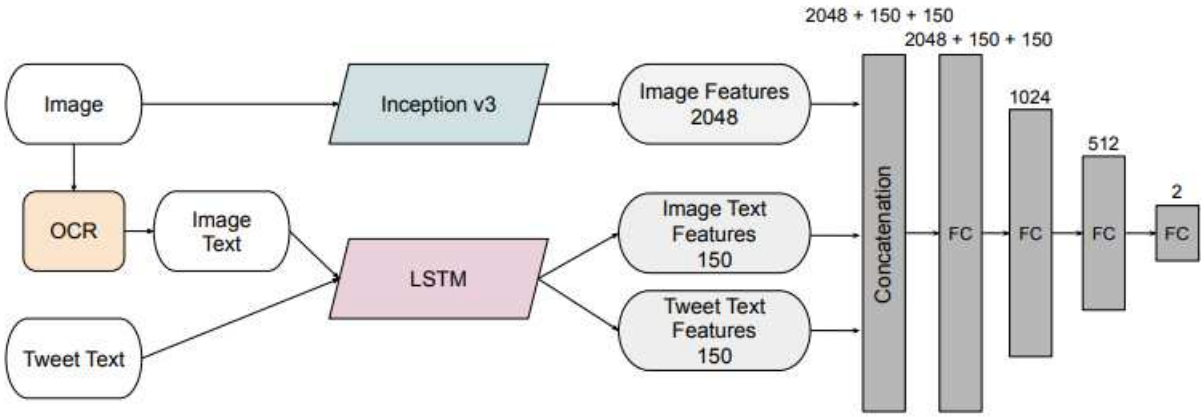


Fonte: o próprio autor.

5.3.2 Investigação de abordagens multimodais para lidar com conteúdo impróprio

Nesta etapa, o objetivo era investigar maneiras de lidar com conteúdos ofensivos. O estagiário ficou responsável pela busca, documentação e experimentações das abordagens mais promissoras. Nesse contexto, algumas abordagens mostraram-se bastante promissoras. Em uma delas, proposta por (GOMEZ et al., 2020), há a coleta e disponibilização do conjunto de dados MMHS150k [4], que contém cento e cinquenta mil exemplos de treinamento de *tweets* ofensivos. Além disso, a arquitetura com melhor desempenho na tarefa, que pode ser ilustrada na Figura 6, foi reportada como uma possível abordagem para este tipo de problema. Nela, o texto extraído da imagem é utilizado, juntamente com o texto do *tweet*, como entrada para uma rede LSTM (*Long Short Term Memory*), e as *features* textuais são concatenadas às visuais aplicando-se uma redução de dimensionalidade gradual.

Figura 6 – *Pipeline* do modelo de concatenação.



Fonte: (GOMEZ et al., 2020)

A reprodução dos resultados não foi possível porque o repositório da implementação não foi disponibilizado. No entanto, a abordagem e a busca por outros conjuntos de dados, como o NSFW Data Scraper [5] e o conjunto de dados NSFW data source URLs [6], foram úteis para fomentar discussões do tema por parte do estagiário com o *Scrum Team*.

Grande parte das abordagens envolve o alinhamento de modalidades por meio de espaços de representação vetorial. O problema deste tipo de abordagem é que é difícil

[4] Multimodal Hate Speech Dataset. Acessado em: 09 de Maio de 2021. Disponível em: https://gombru.github.io/2019/10/09/MMHS/

[5] NSFW Data Scraper. Acessado em: 09 de Maio de 2021. Disponível em: https://github.com/alex000kim/nsfw$_data_s$craper

[6] NSFW data source URLs. Acessado em: 09 de Maio de 2021. Disponível em: https://github.com/alex000kim/nsfw$_data_s$craper

determinar o contexto cultural e semântico entre as combinações que tornam o conteúdo ofensivo ou não.

Seguindo a abordagem proposta por (ABHISHEK D., 2014), foram realizadas experimentações com os modelos BERT, ResNeXt, Image Captioning. Nesta linha de trabalho, utiliza-se modelos pré-treinados de *image captioning*, que geram uma descrição textual de uma figura, além de *Transformers SOTA*, capazes de explorar a análise de sentimento das modalidades para compreensão do contexto existente entre elas.

A documentação das implementações é insuficiente e não descreve de forma completa o preprocessamento aplicado ao conjunto de dados *Facebook Hateful Memes Challenge* [7]. A estrutura dos *scripts* de treinamento não é clara, não sendo possível prosseguir com o processo de experimentação. Houve a tentativa de contato com o autor do artigo e não obteve-se resposta. Isso levou o estagiário a buscar, posteriormente, por soluções com melhores documentações.

5.3.3 Investigação de ferramentas de código-aberto para reconhecimento ótico de caracteres

OCR (*Optical Character Recognition*) é uma tecnologia que permite gerar um arquivo de texto editável por um computador a partir do reconhecimento de caracteres em uma imagem. Existem APIs OCR de alto desempenho, como o Tesseract [8] da Google. No entanto, é uma ferramenta paga que impõe inúmeras restrições comerciais. O estagiário ficou, então, responsável pela busca, teste e avaliação de ferramentas OCR de código-aberto. Esta tarefa era de grande importância para o domínio do problema, visto que a maior parte dos conteúdos multimodais são imagens contendo texto, às quais é necessário aplicar a extração do conteúdo textual em alguma etapa do *Pipeline*.

A partir das indicações feitas nos repositórios das APIs Calamari OCR [9] e Easy OCR [10], obteve-se o conjunto de dados UW3, que consiste em páginas em Inglês moderno com rótulos relativos ao resultado de extração esperado. O estagiário também desenvolveu um gerador de imagens 200x50px contendo frases aleatórias em inglês. O objetivo deste gerador era obter um *Ground Truth* para avaliar as APIs de OCR de forma objetiva utilizando métricas já consolidadas.

Os experimentos exibidos na Tabela 3 demonstraram a capacidade do Calamari em tarefas de OCR. A performance dessa API foi superior à ferramenta Tesseract, tanto

[7] Hateful Memes Challenge and Dataset | Facebook. Acessado em: 08 de Maio de 2021. Disponível em: https://ai.facebook.com/tools/hatefulmemes/

[8] Tesseract OCR | Google. Acessado em: 08 de Maio de 2021. Disponível em: https://opensource.google/projects/tesseract

[9] Calamari OCR. Acessado em 08 de Maio de 2021. Disponível em: https://github.com/Calamari-OCR/calamari

[10] EasyOCR. Acessado em: 08 de Maio de 2021. Disponível em: https://github.com/JaidedAI/EasyOCR

em termos da taxa CER (*Character Error Rate*), quanto no quesito tempo de predição. A redução significativa no tempo de predição pode ser atribuída à utilização do Tensorflow como *framework* principal.

Tabela 3 – Resultados comparativos para o conjunto de dados UW3.

OCR API	**Conjunto de Dados**	**CER**	**Tempo de Predição**
Calamari	UW3	0,155%	3ms
Tesseract	UW3	0,397%	550ms

Como pode-se observar na Tabela 4, o Tesseract obteve um desempenho melhor com alfabetos, enquanto que o EasyOCR teve melhores resultados com números. Além disso, vale-se salientar que as saídas obtidas com o EasyOCR são em caixa-baixa, limitando o seu uso a problemas onde letras maiúsculas não são relevantes. No caso de uma solução onde busca-se um desempenho satisfatório para letras e números, uma abordagem híbrida pode ser aplicada. No quesito tempo de reconhecimento, o EasyOCR se beneficia a partir do uso de uma GPU (*Graphic Processing Unit*), enquanto que o Tesseract é mais rápido em CPUs.

Tabela 4 – Resultados comparativos para o conjunto de dados gerado aleatoriamente.

OCR API	**Conjunto de Dados**	**Taxa de Erro em Números**	**Taxa de Erro em Letras**	**Tempo Usando CPU**	**Tempo Usando GPU**
Tesseract	Randomly Generated	5.5%	0.7%	0.3s	0.25s
EasyOCR	Randomly Generated	1.9%	4.3%	0.82s	0.07s

5.3.4 Investigação de abordagens utilizando *Transformers* VL

A partir das *Sprint Reviews*, as atividades foram direcionadas para exploração de abordagens que não só utilizassem arquiteturas comuns de classificação multimodal, mas que, também, fossem capazes de explorar as relações sutis entre imagem e texto. A partir da leitura de alguns trabalhos da área, constatou-se que incorporar *Transformers* de fluxo simples, que processam uma única modalidade, combinados com modelos de fluxo duplo, que processam duas modalidades, podem resultar em predições mais precisas para conteúdos multimodais.

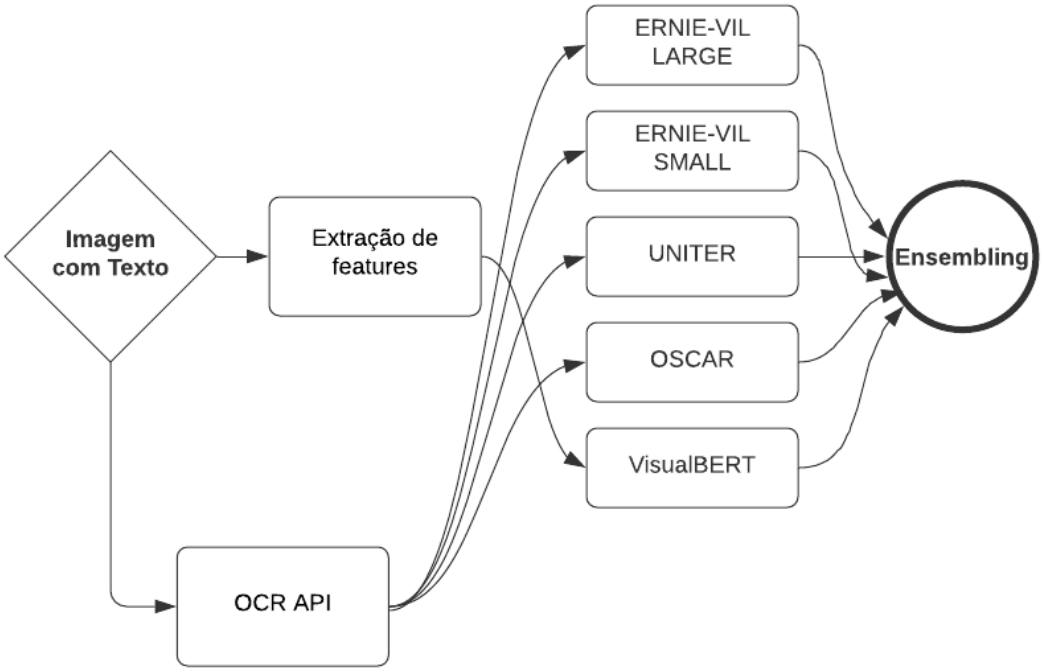
Para avaliar isto, o estagiário ficou responsável por investigar a forma como os modelos e implementações mais bem colocados na competição *Hateful Memes Challenge* [11] funcionavam. Esta investigação era importante pois compreendia soluções estado da

[11] Hateful Memes Challenge. Acessado em: 08 de Maio de 2021. Disponível em: https://github.com/drivendataorg/hateful-memes

arte no contexto de aprendizagem multimodal.

O estagiário iniciou a exploração da solução vice-campeã da competição. O trabalho proposto por (ZHANG et al., 2020) apresenta o Vilio, uma implementação de modelos VL aplicados ao conjunto de dados *Hateful Memes Dataset*. Nele há a utilização de cinco modelos diferentes, utilizando métodos de junção na composição da predição final. O *pipeline* completo da solução pode ser visualizado na Figura 7.

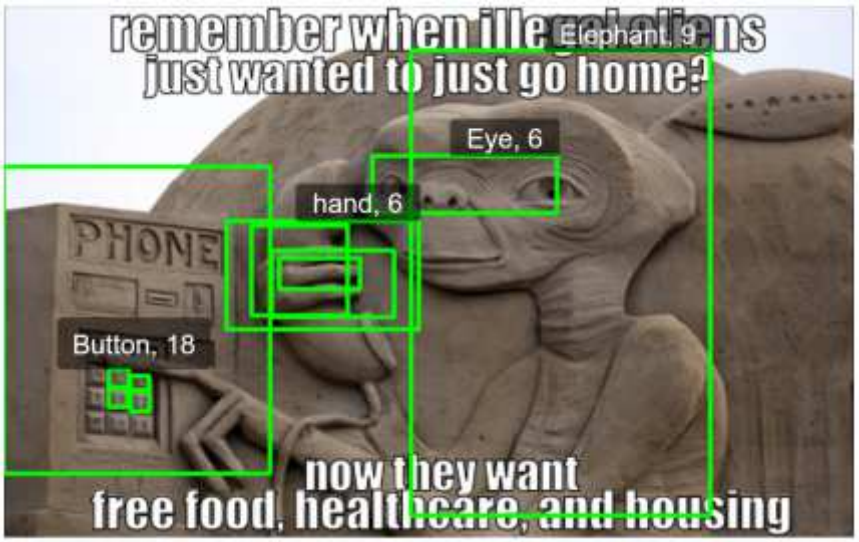
Figura 7 – *Pipeline completo.*



Fonte: adaptado de (ZHANG et al., 2020)

A primeira etapa consistiu na extração das *features* visuais presentes na imagem utilizando o *framework detectron*2. Ao utilizar *features* ao invés de imagens completas, o processo de treinamento é acelerado, além de utilizar somente o que é mais relevante na imagem. Um exemplo dessa extração pode ser visualizado na Figura 8.

Figura 8 – Exemplo de detecção de *features* em uma imagem de treinamento.



Fonte:(ZHANG et al., 2020)

Para realizar a inferência em exemplos inéditos a partir dos modelos pré-treinados, utilizou-se a ferramenta *Kaggle Kernel* [12], um ambiente de desenvolvimento que disponibiliza o acesso a GPUs. O estagiário configurou três arquivos que dividem o *pipeline* de inferência, de tal forma que o custo computacional fosse fracionado. As saídas das duas primeiras etapas de inferência foram utilizadas para a terceira etapa no formato de arquivos .csv. Cada etapa tem duração estimada de duas horas.

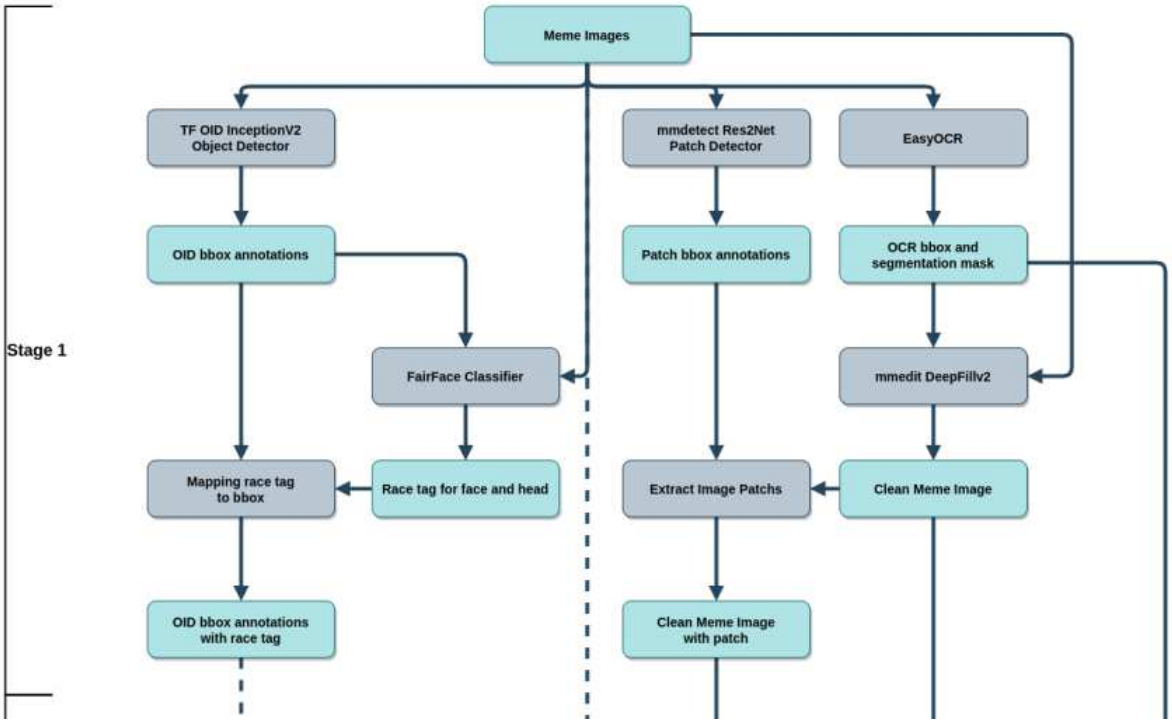
Apesar de todo o processo satisfazer os requisitos necessários, não foi possível completar a etapa de inferência. A quantidade de arquivos (137GB) de entrada para replicar as inferências realizadas e descritas no artigo excedia o limite de armazenamento da ferramenta de computação em nuvem, o que ocasionou erros de reinicialização do ambiente de desenvolvimento. Diante desses problemas, o estagiário foi direcionado a replicar os experimentos descritos pela implementação campeã da competição antes de utilizar uma máquina com GPU a ser disponibilizada.

A solução proposta por (ZHU, 2020) consiste em uma arquitetura composta por quatro modelos VL: VL-BERT, UNITER, VILLA, e ERNIE-Vil. Propõe-se modificações a todos os modelos, exceto para o ERNIE-Vil. O que esta abordagem agrega ao cenário atual é propor a utilização do classificador *FairFace* e do *Entity Detection* da Google. Estes são utilizadas para identificar os contextos históricos e culturais associados aos elementos presentes na imagem.

Os dois estágios do *Pipeline* propostos por esta abordagem e utilizados para as experimentações podem ser observados na Figura 9 e na Figura 10.

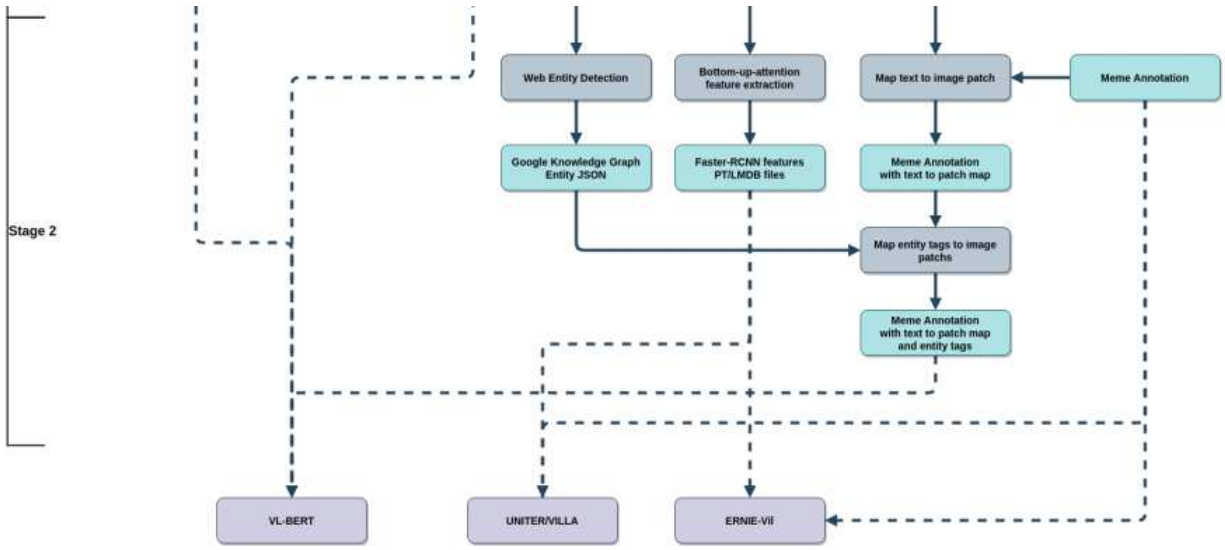
[12] Kaggle Kernel Code. Acesso em 08 de Maio de 2021. Disponível em: https://www.kaggle.com/code

Figura 9 – *Pipeline* - estágio I.



Fonte: (ZHU, 2020)

Figura 10 – *Pipeline* - estágio II.



Fonte: (ZHU, 2020)

A primeira etapa na experimentação consistiu na junção dos modelos pré-treinados. Inicialmente a documentação orienta para a execução de um dos *script shell* utilizando

Docker. No entanto, a ferramenta Google Colab não tem suporte para Docker. A solução inicial foi a replicação manual das etapas envolvidas no *script*.

As dependências associadas ao arquivo de *ensembling* foram configuradas manualmente, já que não havia nenhum arquivo de requisitos que fizesse essa configuração de forma automática. No entanto, houve problemas ao configurar os arquivos de *Pipeline* de NLP de língua inglesa 'encore_web_lg'. A alternativa foi a utilização de outro *pipeline* pré-treinado, o 'encore_web_sm'.

Com a resolução destes problemas, mediante análise de código, verificou-se que o conjunto de modelos pré-treinados não estava sendo passado como argumento para uma das funções que disparava a execução do *pipeline*.

A solução recomendada ao estagiário foi a reexecução do pré-processamento dos dados e da extração das *features*. Inicialmente, os arquivos relacionados ao conjunto de dados *Hateful Memes Challenge* foram alocados a uma pasta específica do diretório presente na documentação [13], bem como os modelos pré-treinados indicados na documentação. Para economizar o tempo de treinamento associado aos modelos, parte dos dados com anotações e as *features* visuais já extraídas foram configuradas para os modelos. Todo esse processo era completamente manual, o que demandava tempo. O estagiário ficou responsável pela automatização no processo de obtenção dos modelos pré-treinados e dados pré-processados, o que contribui para a fluidez de futuros experimentos.

Após a organização dos diretórios, a documentação recomendava a execução de um *Shell script* responsável pela execução da API de OCR, reconstrução da imagem e identificação de *features* visuais significativas. No entanto, o *script* relacionado não está presente no repositório. Como a forma de execução não é explícita, a ausência deste arquivo comprometeu a finalização da etapa de pré-processamento durante o período de atividades do estágio.

[13] Data Preprocessing. Disponível em: https://github.com/HimariO/HatefulMemesChallenge/blob/main/data_utils/ Acesso em: 18 de Maio de 2021.

6 Considerações Finais

A execução desta componente curricular trouxe ao estagiário, e ainda traz, dada a continuidade das atividades do projeto, uma experiência profissionalmente enriquecedora. A partir das atividades realizadas e do envolvimento com o ambiente deste projeto, foi possível a aplicação na prática de vários conceitos vistos durante as disciplinas teóricas do curso, como Informática Industrial, Técnicas de Programação e Gerenciamento, Planejamento e Controle da Produção.

Também foi possibilitado ao aluno o contato com ferramentas que normalmente não são utilizadas nas disciplinas da grade curricular, agregando uma experiência prática e permitindo o contato com soluções estado da arte no campo da aprendizagem profunda.

Ademais, o acompanhamento das etapas práticas envolvidas na execução de um projeto para desenvolvimento ágil de novas soluções, a necessidade de cumprimento de prazos, reuniões, discussões técnicas com profissionais da área e dificuldades encontradas nas atividades estimulam o desenvolvimento de competências e maturidade necessárias para a inserção no mercado de trabalho.

Referências

ABHISHEK D., W. J. S. L. S. Detecting hate speech in multi-modal memes. *arXiv preprint arXiv:2012.14891*, 2014. Citado na página 14.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. Citado na página 6.

BALTRUŠAITIS, T.; AHUJA, C.; MORENCY, L.-P. Multimodal machine learning: A survey and taxonomy. *arXiv preprint arXiv:1705.09406*, 2017. Citado 2 vezes nas páginas 6 e 7.

CHOLLET, F. *Keras documentation: Introduction to Keras for Researchers*. 2020. Disponível em: <https://keras.io/getting_started/intro_to_keras_for_researchers>. Citado na página 9.

DUKE, B.; TAYLOR, G. W. Generalized hadamard-product fusion operators for visual question answering. *2018 15th Conference on Computer and Robot Vision (CRV)*, IEEE, maio 2018. Citado na página 7.

GALLO I., C. A. N. S.; JANJUA, M. K. Image and encoded text fusion for multimodal classification. 2019. Citado 2 vezes nas páginas 6 e 10.

GOMEZ, R. et al. Exploring hate speech detection in multimodal publications. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, mar. 2020. Citado na página 13.

PYTORCH. *PyTorch documentation¶*. 2020. Disponível em: <https://pytorch.org/docs/stable/index.html>. Citado na página 9.

SCHWABER, K.; SUTHERLAND, J. The scrum guide. 2020. Citado na página 3.

SETH, S.; BISWAS, S. Multimodal spam classification using deep learning techniques. *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, IEEE, dez. 2017. Citado na página 11.

TENSORFLOW. *Introduction to TensorFlow*. 2020. Disponível em: <https://www.tensorflow.org/learn>. Citado na página 9.

ZHANG, S. et al. DeVLBert. *Proceedings of the 28th ACM International Conference on Multimedia*, ACM, out. 2020. Citado 2 vezes nas páginas 16 e 17.

ZHU, R. Enhance multimodal transformer with external label and in-domain pretrain: Hateful meme challenge winning solution. ACM, dez. 2020. Citado 2 vezes nas páginas 17 e 18.