Tópicos Avançados em Visão Computacional

Leonardo Rossi Luiz
Pedro Dias Batista
Jorge Augusto Salgado Salhani

I. Introdução

Diversos termos vinculados às áreas científicas são utilizados ao longo do tempo por grandes veículos de mídia como forma de direcionar a atenção geral para determinado avanço tecnológico, em particular àqueles apresentam maior potencial de mudança de paradigma na sociedade. Ao fim dos anos 1970, na era pós moderna,[9] eventos como o lançamento de computadores pessoais [4] e smartphones [1] são notoriamente característicos pela cobertura jornalística massiva e especulações sobre os seus efeitos na sociedade. Até mesmo a invenção do mouse, que constitui periférico cotidiano para usuários de computadores, advém de idealizações de interfaces usuário-máquina que pudessem aprimorar o intelecto humano. [12]

Atualmente o conceito de aprendizado de máquina e redes neurais têm ganhado espaço em noticiários, em particular devido ao processamento de linguagem realizado por modelos como o ChatGPT, da OpenAI, [18] e na sua utilização como ferramenta central para o funcionamento de veículos autônomos, por meio de detecção de objetos e de segmentação semântica (quais pixels da imagem capturada pela câmera do carro constituem uma pessoa, por exemplo) para reconhecimento simultâneo e em tempo real de estradas, pessoas e placas de trânsito, por exemplo [15].

Vale ressaltar que, embora não tão amplamente divulgadas quanto em contextos mencionados acima, estas mesmas tecnologias são fundamentais para diversas outras áreas. Podemos citar o reconhecimento facial como biometria, [10] sistemas de recomendações de conteúdos [19], tradutores de texto [2] e detecção de doenças, como câncer de pulmão [14].

Com base nesse contexto, neste trabalho[1] apresentamos o desenvolvimento de redes neurais sob a perspectiva do campo da visão computacional [11] na realização de tarefas de geração de legendas, também denominado Image Captioning [7]. As seções serão apresentadas como segue: em **Baseline e dataset** apresentamos os modelos que serão utilizados como base para a construção das primeiras versões da nossa rede neural, assim como as bases de dados que utilizaremos para treinamento e teste; em **Resultados iniciais** explicamos os resultados obtidos na implementação deste primeiro modelo, com destaque às estruturas da rede que serão modificados em análises futuras; subsequentemente, em **Resultados aprimorados** destacamos os ganhos e perdas obtidos com as alterações do modelo de base utilizado; por fim, em **Conclusão** retomamos os destaques ao longo do projeto e motivações para novos estudos.

II. Baseline e dataset

Algumas das abordagens propostas para a resolução de problemas vinculados à rotulação automática (captioning)

[1] Código disponível em: https://github.com/jorgesalhani/TopicsVisComp

de imagens são construídas a partir de modelos de machine learning (ML) tradicionais, com extração de atributos (features) e subsequente classificação (por exemplo, por meio de support vector machines - SVM), ou utilizando redes neurais convolucionais (convolutional neural networks - CNN) [7, 16]. Uma vez que o uso de CNNs para a resolução de problemas desta categoria é frequente, assim como sua ampla utilização para demais campos vinculados à classificação de imagens, optamos por utilizar deste modelo para a nossa proposta.

Em resumo, CNNs são constituídas de cadeias de processamentos realizados sobre uma amostra de imagens. Como as componentes responsáveis por cada etapa de processamento são essencialmente funções, seu reposicionamento ou adição de novas camadas (i.e. novas componentes) é bastante flexível, embora não completamente livre. Algumas das camadas mais utilizadas são convoluções, subamostragem (pooling), funções de ativação e camadas densas [11].

Após a extração de features de uma imagem por meio de CNNs, é necessário que um texto com coerência semântica possa ser produzido em função da imagem analisada. Nesta parte, geralmente são utilizados modelos de redes neurais recorrentes (Recurrent Neural Networks - RNNs), em particular arquiteturas com memória longa chamadas Long Short Term Memory (LSTM) [7, 8, 16].

Para este trabalho, consideramos apenas um uma arquitetura de modelo de treino fixa para processamento das featuers e captions. A 1 mostra o modelo utilizado. À esquerda temos o modelo responsável pelo processamento de texto das captions, que recebe as sentenças com input do tamanho máximo observado nos dados da caption. Ao final, esses dados são processados e condensados a um layer de tamanho 256 (defino pelo algoritmo). Já à direita, temos o processamento das features da imagem. Seu input depende do modelo utilizado para extração de características (no exemplo da foto foi usada a VGG16, por isso um input de 4096). De forma análoga ao lado oposto, estes dados são processados e condensados em um layer de tamanho 256. Os dois lados finalizados, essas informações são somadas e processadas juntas. No final, temos uma saída de tamanho do vocabulário das captions para gerar os pesos de predição.

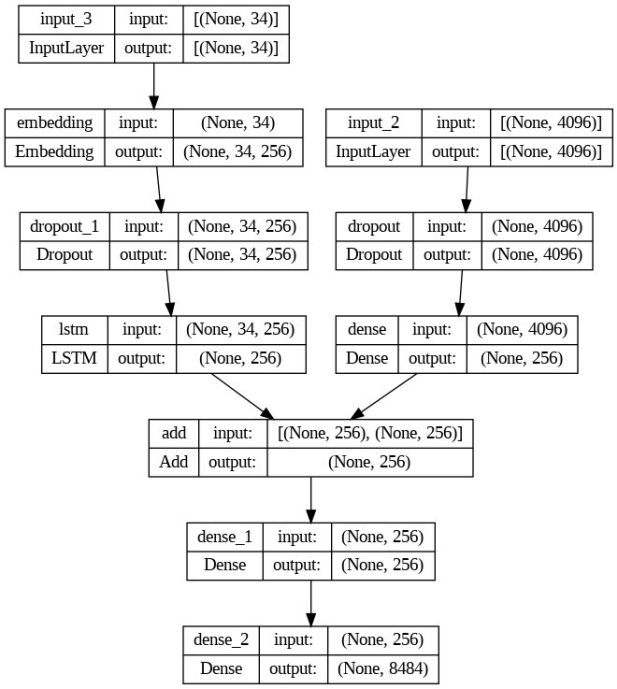

Figura 1: Arquitetura utilizada para o processamento das features (à direita) e captions (à esquerda), unificando ao final para predição de sentença dada imagens de entrada.

Como descrito anteriormente, para o baseline optamos pelas arquiteturas VGGNet16[2] e ResNetV2[3] como pré-treino utilizando o dataset ImageNet[4]. Já para testes, utilizamos o Flickr8k[5] [6, 8]

O FLickr8k é um dataset composto de imagens e respectivos captions. Cada imagem tem um nome único que representa um id e os captions recebem a mesma nomenclatura. Cada imagem possui 5 diferentes captions,

[2]Disponível em: https://keras.io/api/applications/vgg/
[3]Disponível em: https://keras.io/api/applications/resnet/
[4]Disponível em: https://www.image-net.org/
[5]Disponível em: https://shannon.cs.illinois.edu/DenotationGraph/

sendo variações da descrição da imagem, o que leva a uma flexibilidade maior do problema, visto que o modelo não precisa buscar por uma única solução exata. No dataset de captions, a formação segue nome da imagem, seguido de '#' e o número da caption, variando de 0 a 4. Logo após um tab, seguido da descrição. Não existe nenhum tipo de data aumentation nas imagens, sendo todas únicas e de alta variabilidade.

Figura 2: Exemplo de imagem presente no dataset Flickr30k, versão estendida do Flickr8k. Vinculado à imagem, temos as seguintes 5 legendas: ['Crowds of people are walking are multicolored tents and flags that are put up outside in a cement lot as a big city looms in the distance .'] ['Groups of people are in a urban park or walking towards the main area .'] ['People enjoying themselves outDoors , maybe a fair is going on .'] ['People are gathering for some event in the city .'] ['A busy promenade where people gather .']

Vamos considerar em primeira análise a utilização da arquitetura VGG16 como suporte para análise de imagens. A VGG16 é constituída por 1 camada de input de dimensão 224x224x3 (imagens de dimensão 224x224 com 3 cores - RGB), 5 camadas convolucionais, sucedidas, uma a uma, por uma camada de max pooling, e ao topo, 1 camada de flatten, 2 camadas densas e uma camada para predição via regressão logística softmax. [13]

Em resumo, entendemos os processos de cada camada da seguinte forma: camadas convolucionais são responsáveis pela aplicação de uma matriz de filtragem (kernel) sobre a imagem a ser processada, produzindo novas "imagens"(feature maps) cujos pixels resultam da combinação linear de seus pixels vizinhos próximos; camadas de pooling também operam via convolução, porém com objetivo de reduzir a dimensionalidade das matrizes de input. O caso particular de maxpooling seleciona o elemento de maior valor dentre aqueles presentes na região delimitada pelo kernel; camadas de flatten vetorizam (colapsam a matriz 2D em um vetor 1D) as "imagens"de input; camadas densas (fully connected) aplicam funções com pesos e vieses sobre todo o vetor de input, geram valores escalares que podem ser compreendidos, por fim, como elementos de distribuições de probabilidade, quando em conjunto a funções desta natureza, como é o caso de regressões logísticas, tal como a softmax.[11] Para a arquitetura utilizada, as dimensão de uma imagem transformam-se na seguinte sequência:

1) input: (224, 224, 3) - Imagem original
2) convolução 1: (224, 224, 64) - 64 filtros, resultando em 64 feature maps
3) pooling 1: (112, 112, 64) - maxpooling com kernel 2x2
4) convolução 2: (112, 112, 128)
5) pooling 2: (56, 56, 128)
6) convolução 3: (56, 56, 256)
7) pooling 3: (28, 28, 256)
8) convolução 4: (28, 28, 512)
9) pooling 4: (14, 14, 512)
10) convolução 5: (14, 14, 512)
11) pooling 5: (7, 7, 512)
12) flatten: (25088) - colapso da matriz 7x7x512 em um vetor
13) densa 1: (4096)
14) densa 2: (4096)
15) output: (n) - número de categorias a serem preditas

Como desejamos utilizar da arquitetura VGG16 apenas para extração de características das imagens (processamento de feature extraction), em nosso código suprimimos a última camada, ou seja, consideramos como output da CNN o resultado obtido da camada densa 2.

III. Resultados iniciais

Para primeira análise, realizamos a extração de features para um conjunto de 8091 imagens presentes no dataset Flickr8k e separamos 90% das imagens (7281) como conjunto de treino, com os demais 10% para testes (810).

Utilizando apenas 20 épocas (número de vezes nas quais realizamos o treinamento para todo o conjunto de dados delimitados, neste caso 7281 imagens) obtemos um ganho de 3.05 relativo à função de perda (loss function) de entropia cruzada categórica (categorical cross entropy), uma vez que em nossa primeira época obtivemos o valor de 5.22 e, ao fim das 20 épocas, obtivemos 2.17.

Neste momento, vale destacar o uso e importância da loss function como métrica de convergência de resultados. Como visão geral, são chamadas funções de perda (loss functions) funções cujos resultados indicam desvios relativos ao resultado esperado. Como desejamos obter parâmetros e vieses cujos valores apresentam melhores resultados, a cada época de treino calculamos a respectiva loss function. Resultados com menores valores de perda são aqueles cujos parâmetros minimizam a diferença entre valores reais e obtidos. Em particular a entropia cruzada categórica é utilizada para modelos de classificação (discretos, ou também multi-classe), podendo também ser compreendida como uma forma de obter parâmetros de máxima verossimilhança entre os valores das distribuições real e obtida [5].

Podemos também explicitar a distinção da função de perda com a função de ativação, onde utilizamos a função ReLu (Rectified Linear) como substituta à softmax originalmente utilizada na camada de topo (predição) da VGG16.

Funções de ativação são definidas como regularizadores de sinal e são comumente utilizadas em camadas finais de CNNs para que seja possível filtrar valores tais que sua combinação resulte na categoria desejada. Quando utilizamos da função ReLu, valores apresentam valor de 0 ou 1, resultando em matrizes esparsas que podem apresentar ganhos significativos em relação a outras regressões logísticas (softmax ou tanh), embora sejam necessários estudos comparativos para cada função de ativação em uma mesma CNN para garantir qual apresenta melhor performance em dado contexto [3].

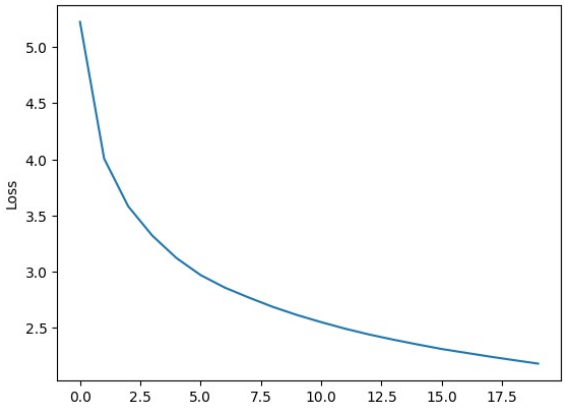

Figura 3: Resultado obtido para a arquitetura VGG16. No gráfico, estão representados os valores obtidos para a função de perda categorical cross-entropy ao longo de 20 épocas. Notamos uma redução significativa de 5.22 (treinamento inicial) a 2.18 (última época considerada).

Com a arquitetura VGG16 mencionada acima, os resultados de treinamento obtidos podem ser observados na figura 3. Nesta figura, fica evidente a tendência de redução da loss function conforme o número de épocas é incrementado. Na figura 4 apresentamos um exemplo de resultado obtido de auto captioning após as 20 épocas consideradas. É evidente a falta de precisão entre imagem e legenda geradas. Algumas hipóteses podem ser consideradas. São elas: pequeno número de épocas consideradas para o

aprendizado e viés de aprendizado, devido à presença de padrões recorrentes no conjunto de teste. Para melhorar nosso resultado, testaremos portanto, variações no número de épocas de treinamento, no dataset utilizado e também para arquiteturas variadas de CNNs.

Figura 4: Exemplo de imagem utilizada para teste de legenda automática. Associada a ela, obtivemos a legenda: ”man in black skirt and woman in black with white shirt are smiling”. Esta associação representa um indicativo de que devem ser consideradas melhoras no modelo, visto que não há relação precisa entre imagem e legenda.

Outra métrica interessante de destaque refere-se à chamada BLEU (Bilingual Evaluation Understudy), cuja utilização faz-se necessária para que exista uma medida objetiva de coesão semântica dos textos de legenda gerados em relação aos originais, sendo 0 relativo a textos não correlatos, e 1 relativo a textos com alta semelhança [17]. Neste trabalho, vamos considerar duas análises do n-grans da BLEU: BLEU-1 com pesos (1, 0, 0, 0) e BLEU-2 com pesos (0.5, 0.5, 0 e 0), sendo o primeiro uma análise ponto a ponto e o segundo uma análise mais de conjunto das sentenças.

Para testes na arquitetura utilizando VGG16, mencionados acima, o valor de BLEU-1 foi de 0.54 e BLEU-2 0.32.

IV. Resultados aprimorados

Após análises feitas observando conjuntos de dados, modelos e resultados em alguns testes, percebemos que uma das maiores dificuldades do contexto abordado remete à complexidade e custo de processamento. Estamos trabalhando com um dataset considerado pequeno para médio para a tarefa de Image Caption. Logo, nossa solução foi tentar gerar aprimoramentos que pudessem manter o poder computacional equilibrado, mas ainda assim obter melhores resultados.

Uma das alternativas para aprimorarmos nossos resultados leva em consideração o número de épocas de treinamento. Na figura 5 estendemos os resultados obtidos anteriormente para 30 épocas adicionais às 20 consideradas na figura 3. Notamos uma redução significativa da loss function e também da métrica BLEU, cujo valor final (apos 50 épocas) foi de BLEU-1 igual a 0.52 e BLEU-2 a 0.29. Para este caso, o aumento de épocas, portanto, não foi suficiente para uma melhora totalmente significativa, porém podemos perceber que ainda não houve uma convergência da loss, nos fazendo crer que um aumento significativo do número de épocas pudesse ajudar no processo, mas isso fugiria da nossa proposta.

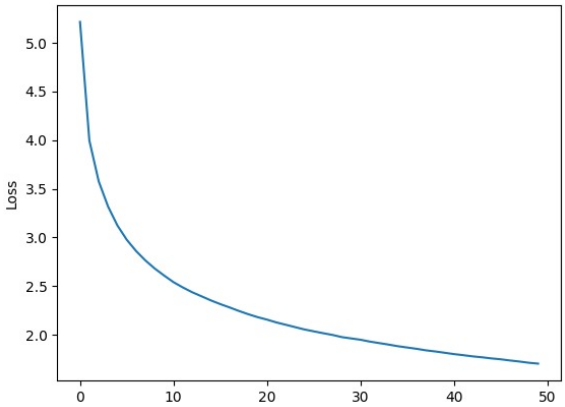

Figura 5: Resultado obtido para a arquitetura VGG16. No gráfico, estão representados os valores obtidos para a função de perda categorical cross-entropy ao longo de 50 épocas. Notamos uma redução significativa de 5.22 (inicial) a 1.70 (última época considerada).

Também podemos realizar uma nova identificação de legenda para a imagem 4. Para esta mesma figura, a nova

legenda descreve ”dog is standing on its back leg and looks at it”. Embora não seja exatamente o que observamos na imagem, é notável o ganho de performance em relação às 20 épocas anteriores, uma vez que houve reconhecimento da figura como um cachorro.

Além do aumento de épocas, podemos também alterar a arquitetura utilizada para extrair as features das imagens. A seguir utilizaremos as arquiteturas InceptionResNetV2[6], EfficientNetB2[7] e ResNet50[8]

Na figura 6 apresentamos os resultados de treino para a função de perda em relação ao número de épocas para extração de features, onde consideramos o máximo de 20 épocas.

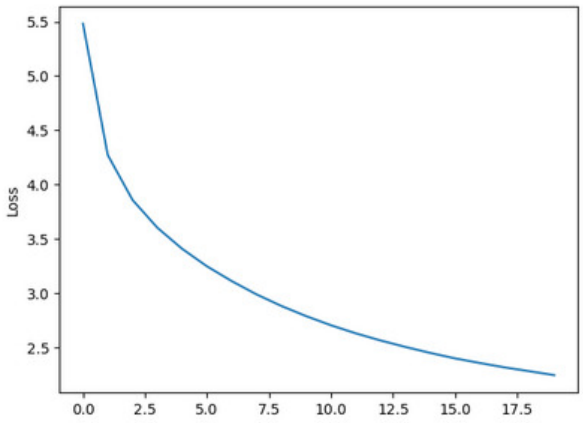

Figura 6: Resultado obtido para a arquitetura EfficientNetB2. No gráfico, estão representados os valores obtidos para a função de perda categorical cross-entropy ao longo de 20 épocas. Notamos uma redução de 5.48 (inicial) a 2.24 (última época considerada).

De modo análogo, nas figuras 7 e 8 apresentamos os resultados para a evolução da loss function ao longo das 20 épocas consideradas para a arquitetura InceptionResNetV2 e ResNet50, respectivamente.

[6]Disponível em: https://keras.io/api/applications/inceptionresnetv2/

[7]Disponível em: https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning/

[8]Disponível em: https://keras.io/api/applications/resnet/#resnet50-function

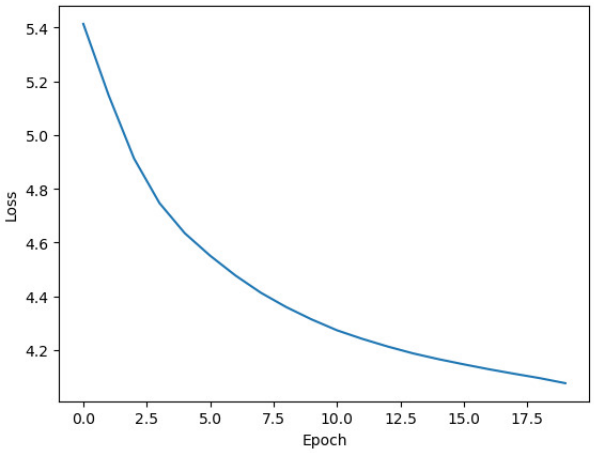

Figura 7: Resultado obtido para a arquitetura InceptionResNetV2. No gráfico, estão representados os valores obtidos para a função de perda categorical cross-entropy ao longo de 20 épocas. Notamos uma redução de 5.41 (inicial) a 4.08 (última época considerada).

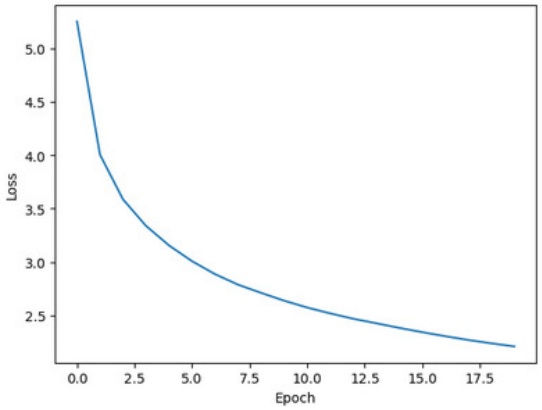

Figura 8: Resultado obtido para a arquitetura ResNet50. No gráfico, estão representados os valores obtidos para a função de perda categorical cross-entropy ao longo de 20 épocas. Notamos uma redução de 5.25 (inicial) a 2.21 (última época considerada).

A seguir, na I, podemos ver os resultados da BLEU para cada um dos modelos de extração de características considerados neste trabalho.

Comparando os resultados para cada arquitetura estudada é possível concluir que o melhor resultado para 20 épocas consideradas foi obtida por meio da arquitetura VGG16

Modelos	BLEU-1	BLEU-2
VGG16 (20)	0.54	0.32
VGG16 (50)	0.52	0.29
InceptionResNetV2	0.51	0.23
EfficientNetB2	0.50	0.26
ResNet50	0.57	0.35

Tabela I: Resultados das métricas BLEU-1 e BLEU-2 obtidos para cada arquitetura utilizada considerando extrações de características ao longo de 20 épocas.

e ResNet50, cujo resultado da função perda foi de 2.18 (vide figura 3). Uma das possíveis justificativas decorre do maior número de features presentes na VGG16 (com 4096) e na ResNet50 (com 2048) quando comparada às demais, que pode dar ao modelo uma maior descrição das imagens, com maiores detalhamentos descriminantes, em particular para problemas cujo número de elementos no dataset é menor.

V. Conclusão

Neste trabalho foi possível aprofundar nossos estudos teóricos e práticos associados ao contexto de redes convolucionais, em particular para tarefas de reconhecimento de imagens. Notamos que algumas arquiteturas apresentam facilidades de implementação e teste, tal como a VGG16, cuja menor profundidade (16 camadas) permite melhor compreensão de cada uma de suas camadas, quando comparada às CNNs InceptionResNet (449 camadas), EfficientNetB2 (186 camadas) e ResNet50 (107 camadas). Vale ressaltar que mesmo as redes mais profundas como a inceptionResNetV2 (449 camadas) não trouxeram os melhores resultados para a predição das legendas, visto que entre as redes analisadas podemos notar que a VGG obteve os melhores resultados.

Outro ponto positivo de análise diz respeito ao processamento de imagens, que ainda é uma tarefa bastante complexa e que exige grande poder computacional. Por mais que os métodos tenham avançado, para tarefas como Caption, de alta complexidade, ainda há muito o que ser aprofundado. Este é um trabalho motivacional de disciplina, o que não nos garante uma gama alta de desenvolvimento devido ao tempo e abordagem, mas em todos nossos testes descritos, implementados e outros não listados neste relatório, podemos com certeza dizer que houve uma grande contribuição na temática para formação.

Finalmente também vale mencionar a importância dos resultados obtidos como incentivo ao estudo mais aprofundado de métricas e modelagens alternativas às CNNs já conhecidas, já que o conhecimento do aparato teórico que envolve cada modelagem matemática é evidente na reestruturação de modelos originais e obtenção de melhores resultados.

Este trabalho contou com o apoio de alguns tutoriais: (1) https://www.youtube.com/watch?v=-cT1m6NZYWc&t=68s; (2) https://www.youtube.com/watch?v=y2BaTt1fxJU&t=220s; (3) https://www.youtube.com/watch?v=fUSTbGrL1tc&t=1s.

Código disponível em: https://github.com/jorgesalhani/TopicsVisComp.

Referências

[1] Ben Agger. ≪iTime: Labor and life in a smartphone era≫. Em: *Time & Society* 20.1 (2011), pp. 119–136. DOI: 10.1177/0961463X10380730. eprint: https://doi.org/10.1177/0961463X10380730. URL: https://doi.org/10.1177/0961463X10380730.

[2] Michael Auli. ≪Joint Language and Translation Modeling with Recurrent Neural Networks≫. Em: *Proc. of EMNLP*. Out. de 2013. URL: https://www.microsoft.com/en-us/research/publication/joint-

language-and-translation-modeling-with-recurrent-neural-networks/.

[3] Bai, Yuhan. ≪RELU-Function and Derived Function Review≫. Em: *SHS Web Conf.* 144 (2022), p. 02006. DOI: 10.1051/shsconf/202214402006. URL: https://doi.org/10.1051/shsconf/202214402006.

[4] Brian Cogan. ≪"Framing usefulness:" An examination of journalistic coverage of the personal computer from 1982–1984≫. Em: *Southern Communication Journal* 70.3 (2005), pp. 248–265. DOI: 10.1080/10417940509373330. eprint: https://doi.org/10.1080/10417940509373330. URL: https://doi.org/10.1080/10417940509373330.

[5] Elliott Gordon-Rodriguez e Gabriel Loaiza-Ganem. *Uses and Abuses of the Cross-Entropy Loss: Case Studies in Modern Deep Learning*. 2020. arXiv: 2011.05231 [stat.ML].

[6] Jiuxiang Gu. ≪An Empirical Study of Language CNN for Image Captioning≫. Em: out. de 2017, pp. 1231–1240. DOI: 10.1109/ICCV.2017.138.

[7] MD. Zakir Hossain. ≪A Comprehensive Survey of Deep Learning for Image Captioning≫. Em: *ACM Comput. Surv.* 51.6 (fev. de 2019). ISSN: 0360-0300. DOI: 10.1145/3295748. URL: https://doi.org/10.1145/3295748.

[8] S. Kalra. ≪Survey of convolutional neural networks for image captioning≫. Em: *Journal of Information and Optimization Sciences* 41.1 (2020), pp. 239–260. DOI: doi:10.1080/02522667.2020.1715602.

[9] Douglas Kellner. ≪Postmodernism as Social Theory: Some Challenges and Problems≫. Em: *Theory, Culture & Society* 5.2-3 (1988), pp. 239–269. DOI: 10.1177/0263276488005002003. eprint: https://doi.org/10.1177/0263276488005002003. URL: https://doi.org/10.1177/0263276488005002003.

[10] Relly Victoria Petrescu. ≪Face Recognition as a Biometric Application≫. Em: *Journal of Mechatronics and Robotics* 3 (2019), pp. 237–257. URL: http://dx.doi.org/10.2139/ssrn.3417325.

[11] Moacir Antonelli Ponti. ≪Everything You Wanted to Know about Deep Learning for Computer Vision but Were Afraid to Ask≫. Em: *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*. 2017, pp. 17–41. DOI: 10.1109/SIBGRAPI-T.2017.12.

[12] Albrecht Schmidt. ≪Augmenting Human Intellect and Amplifying Perception and Cognition≫. Em: *IEEE Pervasive Computing* 16.1 (2017), pp. 6–10. DOI: 10.1109/MPRV.2017.8.

[13] Karen Simonyan e Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[14] Fatma Taher. ≪Lung cancer detection by using artificial neural network and fuzzy clustering methods≫. Em: *2011 IEEE GCC Conference and Exhibition (GCC)*. 2011, pp. 295–298. DOI: 10.1109/IEEEGCC.2011.5752535.

[15] Yu-Ho Tseng. ≪Combination of computer vision detection and segmentation for autonomous driving≫. Em: *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. 2018, pp. 1047–1052. DOI: 10.1109/PLANS.2018.8373485.

[16] H. Wang. ≪An Overview of Image Caption Generation Methods.≫ Em: *Computational intelligence and neuroscience* (2020), pp. 1–13. DOI: https://doi.org/10.1155/2020/3062706.

[17] Krzysztof Wołk e Krzysztof Marasek. *Enhanced Bilingual Evaluation Understudy*. 2015. arXiv: 1509.09088 [cs.CL].

[18] Tianyu Wu. ≪A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development≫. Em: *IEEE/CAA Journal of Automatica*

Sinica 10.5 (2023), pp. 1122–1136. DOI: 10.1109/JAS.2023.123618.

[19] Fei Yu. ≪Network-based recommendation algorithms: A review≫. Em: *Physica A: Statistical Mechanics and its Applications* 452 (2016), pp. 192–208. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2016.02.021. URL: https://www.sciencedirect.com/science/article/pii/S0378437116001874.