IDEA

Innovazione Digitale E Accessibilità per il patrimonio culturale Caproni

TEAM

Ludovico Maria Valenti
Ms Data Science student
ludovico.valenti@studenti.unitn.it

Alessia Meloni
Ms Data Science student
alessia.meloni@studenti.unitn.it

Shaker Mahmud Khandaker
Ms Data Science student
shaker.khandaker@studenti.unitn.it

MENTOR

Paolo Rota
Assistant Professor,
Department of Information Engineering and Computer Science
CIMeC (Center for Mind and Brain)
paolo.rota@unitn.it

ORGANIZERS

Hub Innovazione Trentino
info@trentinoinnovation.eu

CHALLENGE OWNER

Neva Capra
Caproni Project's Representative
Informative System for the Cultural Sector's Representative
neva.capra@provincia.tn.it

USEFUL LINKS

Miro
Github

1. Introduction

Caproni Collection is a part of the provincial cultural heritage which is being studied, protected, preserved and promoted by the Strategic Mission Unit for the Protection and Promotion of Cultural Heritage. The Caproni Collection is a compilation of historic aircraft and engines, historical and artistic assets, archival and library materials in more than 200,000 phototypes. Caproni Collection is considered to be the oldest aviation-themed collection in the world.

The main goal for this challenge, which aims to increase knowledge and accessibility of the cultural heritage to the public, is that of developing an AI solution to automatically take care of two main burdening tasks: subdivide the whole collection according to a thematic criterion; extract the essential and descriptive information (i.e., metadata) from each phototype: main subject, content and textual description. As an additional task, the development of a solution capable of determining the state of conservation of the phototypes.

2. Related works

Literature about the specific context of cultural heritage is scarce, and very few articles mention Artificial Intelligence as a possible solution. Even if most of the time they refer to tangible culture, the focus is on buildings and artifacts, leaving out phototypes. On the contrary, multiple sources, unlinked to the cultural heritage context, have the potential to be successfully applied to it.

Among the articles, one of the most fitted to our scope is that of Wan et al. (2020), where a deep learning approach for the restoration of highly degraded old photos is described. It addresses the generalization problems for complex degraded images and the domain gap between synthetic images and real old photos by training two Variational AutoEncoders. The proposed method achieved improved capability to restore old photos from numerous damages. Even if not directly connected to the tasks we were trying to solve, a module of this model aims to detect the damages.

3. Methods

3.1 Dataset

The dataset we have been provided with is a subset of the Caproni collection. It contains around 9k grayscale medium/high resolution scans of different categories of phototypes such as aircrafts and their components, vehicles, people, landscapes, buildings, factories, figures, objects etc.

In addition, we have manually labeled around 1k Caproni images so that we could use them based on the needs of different tasks. More specifically, we specified the class (or subject), content (different elements in it), caption (general description), and damage level.

Moreover, *COCO*[1], a large-scale object detection, segmentation and captioning dataset has been tailored to perform *semantic segmentation* and accomplish the determination of the phototypes' state of conservation.

3.2 Thematic subdivision

The thematic subdivision task was decided to be addressed as a *classification* problem where each theme corresponds to a class. After looking at the dataset, we decided to define the following classes: airplane, building, factory, figure, landscape, newspaper, object, people, and vehicle.

At the beginning the idea was that of using *Vision Transformers* (Vaswani et al., 2017) models, more precisely the *CLIP* (Contrastive Language-Image-Pre-Training) model (Radford et al., 2021). These allow to classify the images using a zero-shot approach, namely without the need of training and/or fine-tuning models. *CLIP* seemed very promising at the beginning, since it was able to identify an airplane also from the inside. Nevertheless, other tests showed that not all the classes were recognized correctly. These results led us to decide to try other Vision Transformer models and fine-tune them in order to obtain better results.

The chosen models are *vit-base-patch16*, *vit-large-patch32*, and *beit-base-patch16*. In order to fine-tune the models, we labeled some of the Caproni images. More precisely, around 100 images per class were labeled, obtaining at the end 1k pictures. 60% of these were used for the training, 20% for the evaluation, and the remaining 20% for the test. The model with the best results was the *ViT patch 32*, with an accuracy of 0.93 in evaluation, and 0.92 in test. It is important to mention that some of the wrongly classified images are limit-cases, where there is more than one subject, and that for this reason could also be classified differently by distinct individuals. For example, an image depicting a man in an airplane could be classified either as a person or as an airplane.

airplane

landscape

vehicle

Figure 1: Example of predicted classes

3.3 Metadata Extraction

The metadata extraction task corresponds to extracting the subject, the content, and a description of the images. The subject extraction was performed using the model obtained from the thematic subdivision task.

[1] *COCO* test dataset 2015

3.3.1 Content extraction

For what concerns the content, two main options were explored: *object detection* and *multi-label classification*. *Object detection* requires to create bounding-boxes around the objects, and save the information about their position and their content in order to train and fine-tune a model. Since this option was unfeasible, we decided to try with a zero-shot approach using *OWL-ViT* (Minderer et al. 2022) and *DETR* (Carion et al. 2020), but the results were not satisfying.

Multi-label classification requires only to specify the labels in order to proceed with the training and fine-tuning. For this reason, a file containing different information about the images already used for the thematic subdivision task was created. It contains the path of the image, the subject, the content, and a short and general description, the caption.

Here too, we decided to use Vision Transformers models. More precisely, we fine-tuned and tested the *vit-base-patch16*, *vit-large-patch-16*, *vit-large-patch32*, *and beit-base-patch16*, *swin-patch4-window7-224*. The best results were reached using the *Swin patch 4*, with an accuracy of 95% in evaluation and 93% in test. By inspecting the single-class accuracy, it can be noticed that figures, newspapers, factories and vehicles are correctly classified 100% of the time, while objects just 73% of the time. This last poor result could be imputed to the fact that within the object class there are very heterogeneous examples.

3.3.2 Captioning

Concerning the captioning task, different attempts have been made in order to obtain image descriptions. More in detail, we tried with zero-shot captioning using *CLIP prefix captioning* (Mokady et al. 2021), a CLIP model fine-tuned on *COCO* and *Google Conceptual* datasets. The main problem with this approach was that the model does not always recognize the subject of the image, therefore the caption is not accurate. Also, sometimes the descriptions are too "*creative*": e.g. *"the first photo of the workers in a factory in the 50s"*. We also tried to fine-tune the *CLIP* model, as well as the *ViT* with *GPT2* (Radford et al. 2019), *ViT* with *ROBERTA* (Liu et al. 2019), but the results were not satisfactory.

Finally, we found two other models that allow both zero-shot captioning and fine-tuning: *OFA* (Wang et al. 2022), a unified sequence-to-sequence pretrained model, and *nlpconnect-vit-gpt2-image-captioning*.

The first, consists in a *Visual Question Answering* (Anton et al. 2015) process. These kinds of models require as input an image and a question, in our case "*what does the image depict*", and then output an answer, the description. It is very accurate and can recognize complex subjects, such as a sketch of a person with wings. The "object" class is the one with which it has some difficulties. Conversely, it requires a lot of time compared to other methods.

The second model is a *transformer* fine-tuned for image captioning. One of the main advantages of this model is that of being able to produce a general description taking into account all the elements in the image. Also, it is fast in the process, and it is easy to fine-tune. On the other hand, we noticed that sometimes it "sees" things that are not in the picture, for example a scan of a newspaper page with an article is described as "*a newspaper with a picture*

of a person on it". The not always good results could be due to the fact that our labeled dataset contains very few examples.

Considering the pros and cons of all the options, we decided to use OFA, because, beyond the time required to obtain the results, they are more accurate. The images below show some of the captions obtained with this model.

a black and white photo of a pair of headphones

a black and white photo of the cockpit of a plane

an old airplane with a roof on top of it

a black and white photo of the sphinx and the great pyramid of giza

Figure 2: Example of captions with OFA

3.4 Assessment of the phototypes' condition

Among the objectives, we have been working to determine the state of conservation of the phototypes[2]. In order to provide the project owner with a precise measure of the damage, so as to establish an objective measure of it, the task was initially tackled with techniques that would have allowed them to get a measurable result (e.g., counting the pixel of the mask of the damages).

At first, *AD* (*Anomaly Detection*) techniques were considered. Broadly speaking, *AD* consists of detecting anomalies (e.g., damaged images) by learning what should be considered normal (e.g., intact images). This usually happens by ingesting lots of examples of normality, eventually allowing the model to detect what deviates from it. More recent techniques such as *PaDiM*, *PatchCore* and *CFlow-AD* also output a predicted heat map of the detected anomalies. *PatchCore* has been explored and tested, but unfortunately it did not lead to appreciable results. According to the typical datasets these models are trained with, we inferred that the underlying concept of normality greatly differs from the one in our scenario. In detail, among the most famous benchmark datasets for *AD* it can be mentioned *MVTec AD*. It contains multiple objects in an industrial setting (i.e., static background among the different examples), the object of interest is always centered and at most rotated. In such a scenario, and in contrast with our dataset, the images underlie a clear concept of normality and thus detecting deviation from it is relatively easy. For this reason, we dropped *AD* as a feasible solution and moved on with *segmentation*.

[2] From here on, the terms 'picture' and 'phototype' are misleadingly used as a synonym of scan of the original phototype. Since we only had access to the scans of the original pictures, it should be clear that the inferences and applied models refer to the scans rather than the originals.

3.4.1 Segmentation of the damage

Segmentation could be briefly described as a pixel-wise classification, which means that each pixel of an image is assigned to a certain class (e.g., background, scratch, stain, …).

As mentioned in the introduction, the starting dataset (a subset of the Caproni collection) does not contain any annotation about the damages. This fact was forcing us to proceed with an unappealing zero-shot semantic segmentation approach (Bucher et al. 2019, Li et al., 2022), and an unsupervised scenario that was limiting the toolbox of possible techniques at our disposal. For this reason we have decided to implement a custom annotated dataset starting from the images contained in the COCO dataset and from textures of several kinds of damage. The idea behind this choice was that of injecting damages into undamaged pictures, which granted us a ground truth mask of such damages and hence allowed us to switch to a supervised learning scenario.

To be more specific, in its last revision (v4), the dataset we have implemented contains 81,434 unique images taken from *COCO*, gray scaled and resized at 512x512[3]. The damages were taken from different websites providing license-free textures. In this way we managed to gather 553 unique damages ready to be injected in the original *COCO* dataset. Differently from the previous versions of such dataset, the possible classes of damages are: vertical bands, stains and scratches[4]. In order to grant variation, the damages are injected according to certain randomly chosen parameters: placement, size and rotation[5]. Different segmentation models have been tested (*Unet*, *Unet++*, *PAN*, *DeepLabV3*, *DeepLabV3Plus*), along with different encoders pretrained on ImageNet (*ResNet101*, *ResNet152*, *EfficientNet-b7*). Moreover, some transformers have been tried out (*SegFormer*, *MaskFormer* and *DETR*). In all the cases, the models have not been used out-of-the-box, but always fine-tuned with our custom dataset. Among all the tested models, the one providing the best results in validation was *DeepLabV3Plus* with images resized at 224x224, without any image normalization and *ResNet101* as encoder. In this way we were able to obtain 96% mean IoU (counting also the background during the computation). Despite the great results in our validation dataset, also visually appreciable, they did not generalize to the actual Caproni dataset. Since we did not manage to manually annotate the images from Caproni we could not compute a metric, so we determined the best performing model through a visual inspection and comparison of the output masks with some pre-selected images from Caproni. Unfortunately, scratches and small damages were hardly detected. Nonetheless, vertical bands and stains were recognized properly in most of the cases.

---

[3] In the first versions of this dataset, the size was 224x224. Given the amount of images, to reduce the overall time of creation of the dataset, we had to constrain the size of the produced images at first. Still, the process was slow (~5 images per second).

[4] The previous versions also included dots. By inspecting the produced images, we realized that dots and stains were overlapping in their features. For this reason we have decided to merge them in just one class: stains.

[5] Placement regards the chosen x and y coordinates for a given damage. Size regards the chosen size for the damage, while keeping its aspect ratio, and it ranges from 0.6 (decreasing its size) to 1.2 (increasing its size). Rotation regards the rotation of the damage prior to its injection, it ranges from -180° to 180°.

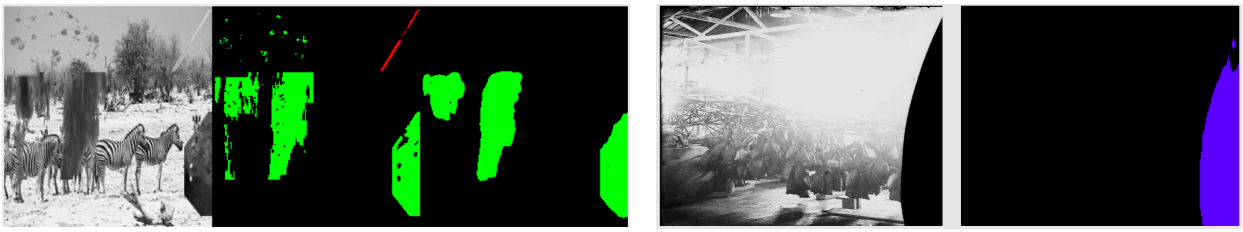


Figure 3: On the left an image from the custom dataset with the ground-truth (center) and predicted mask (right). On the right the generalization of a Caproni image with the predicted mask on its right.

After that, we tested a model taken from the paper *Bringing Old Photos Back to Life (CVPR2020)*, under the promise of restoring old pictures, this model contains multiple sub-modules that allow to restore pictures with and without scratches, detect scratches, perform a global restoration and enhance the faces. Since our model was performing quite well with bands and stains, we decided to implement an *ensemble model* altogether with this one, just by taking its sub-module that detects scratches. In this way, the task was splitted among the two models: bands and stains are detected by our model, while scratches and small defects are detected by Microsoft model. Even if the performances improved overall, the number of false positives increased (i.e., straight lines in the pictures were often confused with scratches), as it is possible to observe in Figure 4.

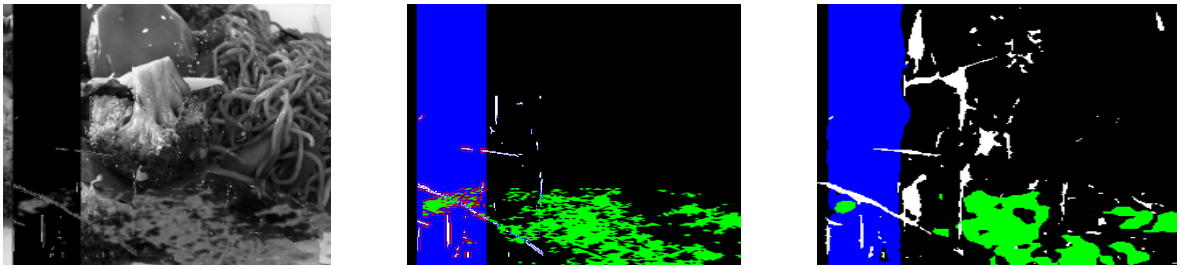


Image with damages *Damage masks* *Predicted masks*

Figure 4: Example of damage segmentation with an ensemble model

3.4.2 Classification of the damage

As a last resort, some *classification* techniques were implemented. In the specific context of computer vision, *classification* consists of assigning a label, among the ones in which the model was trained with, to each image. A subset of the images from Caproni has been splitted in three possible classes: 0 (undamaged, few damages), 1 (some damages, several damages), 2 (completely damaged) according to the observable state of conservation of each given picture.

For classification some transformers have been fine-tuned (*SWIN*, *BEIT*, *DEIT* and *VIT*) in most of their available versions and therefore tested. The best model resulted to be *Swin* (version *large-patch4-window7-224*) providing 84% accuracy. Despite the fact *classification* outperforms all the segmentation techniques we have been tested, its clear limitation is the fact that it is not providing a mask that would instead provide us with a measurable result. *Segmentation*, on the other hand, seems to be a very promising technique, but to work properly

it definitely requires some proper damage annotations so to let the model be applied in the same *domain*. If the manual annotation would result in a burden task, another possibility would be that of exploring *segmentation* techniques in an *unsupervised domain adaptation* scenario. In this case, the model would be trained in the very same dataset we did compose, but, broadly speaking, aligning the two domains (*source*: custom dataset and *target*: Caproni dataset), so to let the model transfer its capability also to Caproni.

no_few_damage

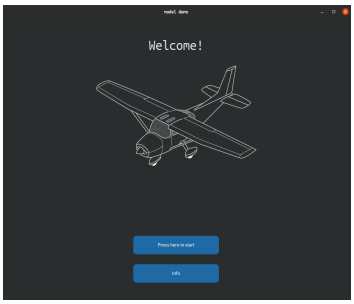
damaged

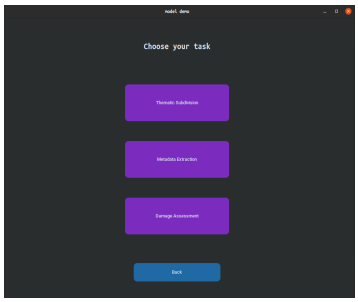
completely_damaged

Figure 5: Example of damage classification

3.5 Desktop Application Development

We integrated all the developed solutions into a dynamic, ready-to-use, and user-friendly desktop application. It can be used for all the tasks mentioned above. It has three possible processes: thematic subdivision, metadata extraction, and damage assessment.





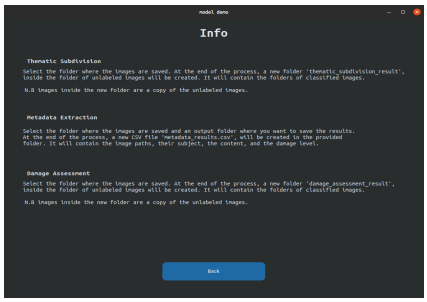


Figure 6: The desktop application

In order to perform the thematic subdivision, the user only needs to select the folder where all the images are stored. At the end of the process, a new folder, containing the classes subfolders, is created. Inside the subfolders, there will be the copies of the classified original images.

In the metadata extraction process, the user has to select the folder where all images are stored, and a folder where he wants to save the results. At the end of the process, a CSV file will be created in the specified folder. It will contain different information about all the images: image path, subject, content, description, and conservation status.

The damage assessment process works as the thematic subdivision one. The user selects the folder where the images are stored, and at the end there will be a new folder with three

sub-folders, one for each damage level. Inside the sub-folders there will be the copies of the original images.

image_path	subject	content	description	damage
FC_21_01331.jpg	vehicle	airplane	an old airplane with a roof on top of it	no_few_damage
FC_26_02290.jpg	building	building, people	black and white photo of the sphinx and the great pyramid of giza	no_few_damage
FC_4_00187.jpg	object	object	a black and white photo of a pair of headphones	no_few_damage
FC_5_00300.jpg	airplane	airplane	a black and white photo of the cockpit of a plane	damaged

Table 1: Example of metadata extraction results

4. Results and discussion

In the following table are summarized the numerical results of our best performative models:

		Model	Metrics	Domain
Thematic Subdivision		*ViT*	Acc: 0.92	*Caproni*
	Subject	*ViT*	Acc: 0.92	Caproni
Metadata	*Content*	*Swin*	Acc: 0.93	Caproni
	Caption	*OFA*	NA	Caproni
Damage Assessment *(segmentation)*		*DeepLabV3Plus*	mIoU: 0.97	Custom Dataset
Damage Assessment *(classification)*		*Swin*	Acc: 0.84	Caproni

Table 2: Summary of models results

Each row represents the task for which the model has been implemented. On the columns the model used, the test values for the computed metrics, and the domain to which the metric refers to. It should be specified that the accuracy has been computed in different ways according to the task, but in general it always counts the number of correctly classified examples over the total number of examples. The domain refers to the dataset in which a given model has been trained, evaluated and tested in: “Caproni” means that the dataset provided by the project owner has been used, “Custom Dataset” means that our custom *COCO* dataset has been used. In the

latter case, we were forced to do so, since, as already mentioned, we had no annotation about the damages that could have been used as ground truth reference. For this reason a quantitative evaluation in Caproni was not feasible. Similarly, for the captioning task we had no ground truth captions, so we could not provide a metric about the model's performance. Nonetheless, in both cases, the evaluation has been made through a qualitative inspection.

Going in details of the obtained results, the chosen model for content extraction is *Swin-large*, which provided an overall accuracy of 0.93. Considering the single class accuracy, *factory*, *figure*, *newspaper*, and *vehicle* are correctly classified 100% of the time. *Airplane*, *building*, and *people* classes have an accuracy above 0.9, while *landscape* and *object* are correctly classified 84% and 73% of the time respectively. The low performance obtained in the last two classes could be due to the fact that both contain very different examples, such as an aerial view of a city and a field, a cabinet and a clock.

The captioning task was the most difficult one to address, that is mainly for two reasons: small labeled dataset, missing of an objective way to determine the performance. In fact, captioning requires a lot of examples to perform well, in particular when there are unseen and domain-specific objects, as it is in our case. In fact, Caproni contains specific artifacts that are difficult to identify also for a non-professional individual and that require a deep knowledge in the aviation field. Nonetheless, if we consider other kinds of images in our dataset, such as pictures depicting airplanes, landscapes, or people, the model performs well.

One of the main advantages of the chosen models, particularly for what concerns the thematic subdivision, is the possibility of an easy generalization. In fact, with respect to more traditional methods that require lots of examples (around 1k per class), these ones require instead very few examples (around 100 per class). This means that the same models can be easily fine-tuned with other datasets, for example one concerning ancient amphoras.

5. Further works

Taking into account the state of the art of the use of Artificial Intelligence in the cultural heritage domain, much more could be done.

Considering the given project, having more resources, such as labeled images, better results can be obtained in the images description and damage assessment tasks. Particularly, having masks of the damages could allow one to develop a more precise model. It will allow us to determine, in an objective way, the deterioration of a given phototype, having the amount of total damage.

Keeping in mind this last point, other further development of this project would include the capability of recognizing individuals (e.g., eng. Caproni), as well as recognizing the specific aircraft models. This would be possible through classification, notwithstanding more detailed annotations of the dataset, or through OCR in the cases in which the IDs of aircrafts are textually specified in the images.

Related to the possibility of obtaining a model able to identify and segment the deteriorated part, there is the potential to virtually restore the phototypes. This would make it possible to

have previously absent or partial information, increasing in this way the value and potential of the cultural asset.

In conclusion, one of the greatest potentials of the use of Artificial Intelligence in this domain lies in its capability to automate repetitive and time-consuming tasks. Thus, providing support to professionals in carrying out their work, allowing them to reduce these tasks to a polishing, and consequently allowing them to improve the overall efficiency and leave them space to more stimulating activities.

References

1. Agrawal., A., Lu., J., Antol, S., Mitchell, M., Zitnick C. L., Batra, D., Parikh, D. (2015). *VQA: Visual Question Answering*. arXiv. https://arxiv.org/abs/1505.00468.
2. Bao, H., Dong, L., Piao, S., Wei, F. (2022). *BEiT: BERT Pre-Training of Image Transformers*. arXiv. https://arxiv.org/abs/2106.08254.
3. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C. (2019). *MVTec AD--A comprehensive real-world dataset for unsupervised anomaly detection*. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9592-9600). https://openaccess.thecvf.com/content_CVPR_2019/papers/Bergmann_MVTec_AD_--_A_Comprehensive_Real-World_Dataset_for_Unsupervised_Anomaly_CVPR_2019_paper.pdf
4. Bucher, M., Vu, T. H., Cord, M., & Pérez, P. (2019). Zero-shot semantic segmentation. Advances in Neural Information Processing Systems, 32.
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S. (2020). *End-to-end object detection with transformers*. arXiv. https://arxiv.org/abs/2005.12872.
6. Chen, L., Papandreou, G., Schroff, F., Adam, H. (2017). *Rethinking Atrous Convolution for Semantic Image Segmentation*. arXiv. https://arxiv.org/abs/1706.05587.
7. Chen, L., Zhu, Y., Papandreou, G., Schroff, F., Adam, H. (2018). *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. arXiv. https://arxiv.org/abs/1802.02611.
8. Cheng, B., Schwing, A. G., Kirillov, A. (2021). *Per-Pixel Classification is Not All You Need for Semantic Segmentation*. arXiv. https://arxiv.org/abs/2107.06278.
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv. https://doi.org/10.48550/arxiv.2010.11929
10. Li, Y., Yao, H., Wang, H., & Li, X. (2022). FreeSeg: Free Mask from Interpretable Contrastive Language-Image Pretraining for Semantic Segmentation. arXiv preprint arXiv:2209.13558. https://doi.org/10.48550/arxiv.2209.13558
11. Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, L., Dollár P. (2015). *Microsoft COCO: Common Objects in Context*. arXiv. https://arxiv.org/abs/1405.0312.
12. Liu Z., Lin Y., Cao Y., Hu H., Wei Y., Zhang Z., Lin S., Guo B. (2021). *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. arXiv. https://arxiv.org/abs/2103.14030.
13. Liu, S., Qi, L., Qin, H., Shi, J., Jia., J. (2018). *Path Aggregation Network for Instance Segmentation*. arXiv. https://arxiv.org/abs/1803.01534.
14. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv. https://arxiv.org/abs/1907.11692.

15. Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., Wang, X., Zhai, X., Kipf, T., Houlsby N. (2022). arXiv. https://arxiv.org/abs/2205.06230.
16. Mokady, R., Hertz, A., & Bermano, A. H. (2021). *Clipcap: Clip prefix for image captioning*. arXiv. https://doi.org/10.48550/arxiv.2111.09734.
17. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I. (2021). *Learning Transferable Visual Models From Natural Language Supervision*. arXiv. https://doi.org/10.48550/arxiv.2103.00020
18. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019). *Language models are unsupervised multitask learners*. OpenAI blog, 1(8), 9. https://openai.com/blog/better-language-models/
19. Ronneberger, O., Fischer, P., Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv. https://arxiv.org/abs/1505.04597.
20. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H. (2021). *Training data-efficient image transformers & distillation through attention*. arXiv. https://arxiv.org/abs/2012.12877
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017). *Attention is all you need. arXiv*. https://arxiv.org/abs/1706.03762.
22. Wan, Z., Zhang, B., Chen, D., Zhang, P., Chen, D., Liao, J., Wen, F. (2020). *Bringing old photos back to life*. arXiv. https://arxiv.org/abs/2004.09484.
23. Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., Yang, H. (2022). *OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework*. arXiv. https://arxiv.org/abs/2202.03052.
24. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez J. M., Luo, P. (2021). *SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers*. arXiv. https://arxiv.org/abs/2105.15203.
25. Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., Liang, J. (2018). *UNet++: A Nested U-Net Architecture for Medical Image Segmentation*. arXiv. https://arxiv.org/abs/1807.10165.