Interactive Deep Model Tuning for Surveillance Review

Alvaro Casado-Coscolla*[1], Carlos Sanchez-Belenguer[2], Erik Wolfart[2], and Vitor Sequeira[2]

[1]*Seidor Italy SRL, Milan, Italy*
[2]*European Commission, Joint Research Centre (JRC), Ispra, Italy*

Video Surveillance is a time-consuming task for nuclear safeguards inspectors, having to review the video footage from hundreds of cameras installed in nuclear facilities worldwide. In recent years, deep learning has proved to obtain outstanding results in common computer vision tasks that are relevant for safeguards inspectors such as object detection or scene understanding. Deep learning models have the potential to significantly improve the video review workflow both by providing automated data analysis and by enabling interactive tools supporting the manual surveillance review.

However, several challenges need to be addressed in the context of nuclear safeguards. Use cases vary greatly between different facility types and therefore deep learning models need to be trained or fine-tuned to each specific task. Labelling large sets of training data for each model is not feasible, as it would require too much effort and cannot be outsourced due to the sensitivity of the data. We propose an interactive workflow using pre-trained models that integrates data review, labelling and model tuning. It minimizes the labelling and training effort and gives the inspector control over the tasks learned by the model.

The paper describes the workflow and underlying model architecture and presents experimental results.

1 Introduction

Video surveillance has become ubiquitous for nuclear safeguards, with cameras installed in hundreds of processing and storage facilities around the globe. The video footage recorded by these cameras plays a critical role in nuclear safeguards, but reviewing it can be a time-consuming and challenging task. This process often requires expert knowledge of the complex processes and environments being monitored.

In order to reduce human efforts involved in the review process, several works have proposed the use of computer vision and machine learning. Deep learning has proven to be a

*Working under contract for European Commission, Joint Research Centre (JRC), Ispra, Italy

great resource for computer vision tasks such as object detection [1, 2], scene understanding and segmentation [3], video summarisation [4, 5], or video classification [6] amongst others.

Frame retrieval for video analysis can help in various applications like security and surveillance, traffic monitoring, and industrial inspection. It involves searching and retrieving similar frames or regions of interest from a large collection of videos. This can be useful for identifying specific patterns or events of interest, such as a particular object, condition or activity, thus providing a more comprehensive understanding of the scene being monitored.

In this work we propose an interactive machine learning pipeline for relevant frame retrieval in the context of video surveillance footage review. Our pipeline features a pretrained deep feature extractor and a support vector machine that can be tuned online by the domain experts. The results from this work show that our method is able to solve several frame retrieval use-cases with high accuracy and almost no false negatives.

1.1 Previous Works

Deep models often require thousands of labelled samples in order to be trained. There are several generalist datasets, such as COCO or ImageNet [7, 8], that represent day-to-day objects. However, in the nuclear safeguards field limited data is available.

Synthetic data generation and simulation [9, 10] are a known solution to overcome data scarcity. While these methods reduce the amount of real data that is required, a small amount of real samples is still needed to model the domain shift between synthetic and real data. In addition, these techniques add the overhead of modelling and simulating the environment and the objects of interest, which comes with a cost (in time and money).

Alternatively, it is possible to focus on reducing the need of real data to the very minimum with the use of few-shot models [11, 12]. In addition, we can use the inspector feedback from past and current reviews in order to improve our model's performance. This can be achieved by using techniques such as interactive labelling [13]. It is a process of adding or correcting labels to the training data by an expert or human annotator in order to improve the accuracy of the model. This can be useful in a variety of situations, specially in fields with low data availability.

1.2 Overview

In this work, we introduce an interactive few-shot machine learning pipeline -illustrated in Figure 1- for video frame retrieval on video footage that can be tuned online during the review process. The backbone of the pipeline consists in a deep feature extractor that is able to encode images into meaningful embeddings that are later used to train a machine learning model (e.g. a support vector machine) along with user labels. Finally, we assess

the performance of the proposed pipeline on real data for video frame retrieval tasks.

The rest of this document is structured as follows: Section 2 describes our proposed pipeline for performing the interactive video review; Section 3 shows the achieved results; finally, Section 4 draws some conclusions and outlines several future tasks.

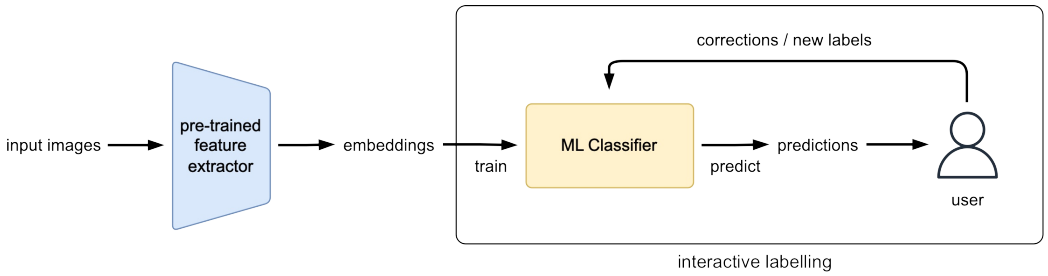


Figure 1: Proposed workflow for interactive training of a machine learning model using a pre-trained deep feature extractor and user feedback.

2 Approach

This work proposes a machine learning pipeline for video frame retrieval in the context of video surveillance footage review. This task consists in finding frames whose contents are similar to the user's query, which can be relevant when the user has found an object of interest or an activity in the video and would like to discover if the same behaviour appears again along the surveillance footage.

Training a deep model for such task requires a huge amount of data (synthetic or real), which may not be feasible in many contexts where not enough time or resources are available for acquiring or generating a proper training dataset. We address this challenge with a machine learning pipeline that uses a pre-trained deep feature extractor in order to obtain highly-representative embeddings of the frame contents that are later used to train a machine learning classifier.

The training stage takes between 10 and 15 minutes per camera when a user is manually annotating the video footage, therefore making it possible to train a model for each camera in a facility in a reasonable amount of time. More importantly, once the model is trained for a use-case it can be used for analysing future campaigns. However, there is always the possibility to re-train it adding new samples from recent campaigns if needed.

2.1 Feature Extraction

Feature extraction is the process of extracting important features or patterns from raw input data, typically images, by passing them through a series of convolutional, pooling, and normalization layers. The goal of this process is to transform the raw pixel information of

an image into a representation that is more useful for the downstream task at hand, such as image classification, detection, or segmentation.

A deep CNN classifier -depicted in Figure 2– typically consists of multiple convolutional and pooling layers that are designed to extract increasingly complex features from an input image. They use a feature extractor whose output is an embedding, a numerical representation of the contents of the input image, which is then passed to the classification head (i.e. a set of fully connected layers) in order to make a final decision on its label).

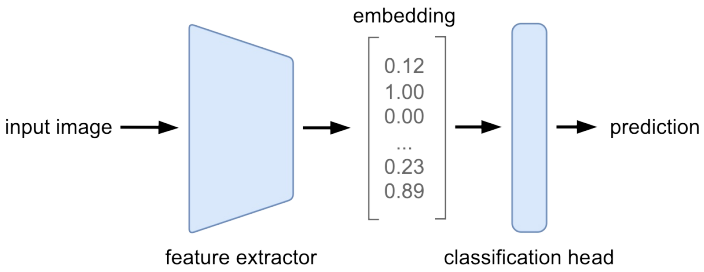


Figure 2: Typical architecture of a deep learning classifier.

In this work, we propose the use of the feature extractor taken from a pre-trained CNN classifier in order to obtain meaningful embeddings without the need of any training. This represents an advantage in scenarios where limited or no labelled data is available, avoiding the need of manually annotating the video footage.

We used OpenAI's CLIP (Contrastive Language-Image Pre-Training) [14] visual feature extractor for this purpose. CLIP is a deep learning model that has been pre-trained on massive amounts of textual and visual data collected from the internet. It is designed to understand the relationship between language and images by mapping both types of input data into the same feature space. This allows to assign similar embeddings to images that contain the same type of object even if their visual appearance is not so similar.

In the proposed pipeline, the user identifies one or more regions of interest (ROIs) in which they observe a relevant object or in which they know some activity will take place. Once ROIs are defined, the deep feature extractor processes all the video frames and generates an embedding for each ROI in each frame. Given that these embeddings contain a numerical representation of the contents ROI, it is possible to compare them with several similarity metrics such as the euclidean distance. However, these similarity metrics may not be sufficient for some use-cases so we decided to treat this task as a learnable classification problem.

2.2 ROI Classification

Each region is treated as a binary classification task in which each frame can be either equal or different. For each region, the user labels an initial frame as a positive example for the

relevant object that they are looking for. For example, an inspector may be looking for all frames that contain a spent fuel cask entering or leaving a reactor pool. In this case, the inspector would select a ROI that contains the area of the pool in the video and a frame in which they can see the cask. Therefore, all video frames that contain a spent fuel cask inside this ROI are considered relevant and the rest of frames become irrelevant for this task.

The extracted embeddings are used to train a support vector machine (SVM). We chose this type of machine learning model because they need a much smaller training dataset compared to deep neural networks, thus reducing the amount of samples that need to be labelled by the user. Given the high-dimensionality of the embeddings it is possible that the decision boundary between similar and not similar frames is not linear. In order to cope with this non-linearity we use a radial basis function.

Another advantage of using SVMs is that they can be trained quickly, thus making it possible to train a SVM after each new label from the user, which allows to update the classification results for all frames in real time. These class predictions (similar or different) are then used to compute a final score that takes into account the predicted class and the distance to the decision boundary of the classified sample. This new score represents the final prediction of our pipeline, assigning higher values to frames that are classified as similar based on the user's input.

3 Results

We evaluated our method on video footage from two different NGSS cameras installed in a nuclear reactor: one camera in the reactor hall and one inside the transfer hatch that connects the reactor to the rest of the facility. Some ROIs were manually defined and annotated frame by frame in order to create a ground-truth dataset for evaluation. In Figure 3 we show an example of an irrelevant and a relevant frame for each of the use-case.

All experiments consist in two stages: first the SVM is trained using a random subset of a labelled dataset for each ROI. This data consists on images from few days of footage that contain both relevant and irrelevant scenes. Afterwards, the trained model was used for evaluating its results on the rest of the labelled data.

For evaluating the model's performance we used several classification metrics. Precision represents how many of our predictions are really relevant frames, while recall -perhaps the most interesting in the safeguards domain- tells us the fraction of relevant frames our model found. Given the amount of true positive (TP), true negative (TN), false negative (FN), and false positive (FP) we can define them as:

$$Acc. = \frac{TP + TN}{TP + TN + FP + FN} \qquad Pre. = \frac{TP}{TP + FP} \qquad Rec. = \frac{TP}{TP + FN}$$

In addition to the frame classification evaluation we also used the same metrics for eval-

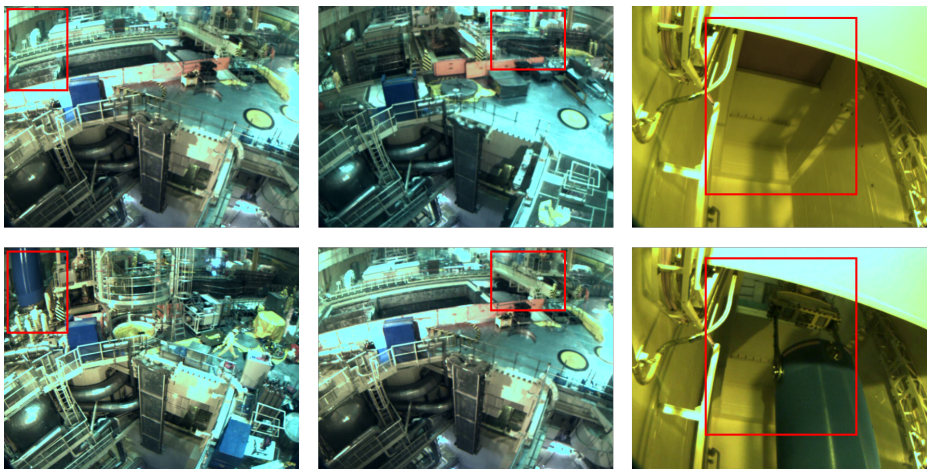


Figure 3: Relevant and irrelevant scenes for the evaluated use-case. From left to right: reactor bridge, spent fuel cask, and hatch. Refer to subsections 3.1-3.3 for more details on the use-cases.

uating the model's performance at event level. An event is a set of consecutive frames from a video in which the ground truth label is the same for all of them (relevant or irrelevant). This allows us to get a better sense of how many relevant events are really missed. For this purpose we define a temporal overlap (TO) threshold that represents the amount of correctly predicted frames are inside an event window for it to be considered correct. For example, if the spent fuel cask appears inside the ROI of the transfer hatch for a total of 20 frames, our model would need to predict as relevant at least 18 frames inside this time period for it to be considered a correct prediction.

For the transfer hatch use-case, an event represents a slice of the video footage in which the cask appears. In the example depicted in Figure 4 there are two events. Our pipeline's predictions are compared to ground-truth at (1) frame level, where we can observe four false negatives and one false positive; and (2) at event level, where we can see that both events were detected correctly since the temporal overlap of the correctly classified frames inside each event is higher than the recommended 70% threshold.

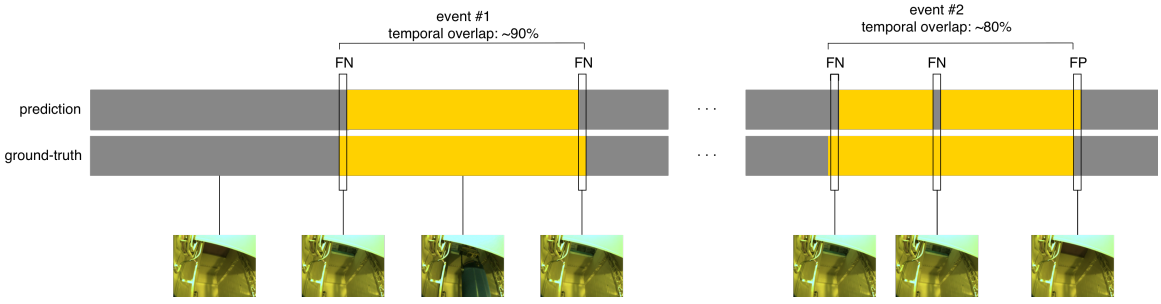


Figure 4: Visual representation of the classification results along the timeline corresponding to video footage from the transfer hatch. Given a set of user-selected labels, frames that are classified as relevant are highlighted with brighter colours, while not relevant frames have darker tones.

3.1 Transfer hatch: open

The transfer hatch represents a critical dataset for confirming the operator declarations. Therefore, finding all frames where any object is being carried by the crane in the hatch can help to validate the campaign. For this use-case the SVM was trained on the data of 6 days of 2021. Then, it was evaluated for the data of 17 days (9 days of 2021 and 8 days of 2022). In Table 1 we can see the average results after running 9 independent experiments using 128 positive and 128 negative labels.

	Accuracy		Precision		Recall	
	μ	σ	μ	σ	μ	σ
Frames	0.9416	0.0973	0.8819	0.1426	0.9998	0.0008
Events @ TO 90%	1.0000	0.0000	0.7782	0.2139	1.0000	0.0000
Events @ TO 70%	1.0000	0.0000	0.9450	0.1200	1.0000	0.0000
Events @ TO 50%	1.0000	0.0000	0.9537	0.0999	1.0000	0.0000

Table 1: Performance results on the transfer hatch dataset.

Our experiments show that the model is able to identify all relevant events (i.e. maximum recall) for all the tested temporal overlap thresholds, while achieving high precision results with a 70% threshold.

3.2 Reactor pool: Spent fuel cask

Another type of important objects of interest in the reactor dataset are spent fuel casks. These casks can be seen entering and leaving the pool while they are being carried by the reactor crane before they are finally transported out of the reactor. For this use-case the SVM was trained on the data of 2 days of 2021 and 2 days of 2022. Then, it was evaluated for the data of 7 days (4 days of 2021 and 3 days of 2022). In Table 2 we can see the average results after running 9 independent experiments using 128 positive and 128 negative labels.

	Accuracy		Precision		Recall	
	μ	σ	μ	σ	μ	σ
Frames	0.9956	0.0031	0.9259	0.0904	0.9264	0.0844
Events @ TO 90%	0.7143	0.4554	0.7143	0.4554	0.7143	0.4554
Events @ TO 70%	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
Events @ TO 50%	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000

Table 2: Performance results on the spent fuel cask dataset.

Our experiments show that the model is able to identify with no errors all relevant events for temporal overlap thresholds 70%, which represents a conservative enough scenario for this use-case.

3.3 Reactor core: Bridge

One of the main objects of interest in the reactor dataset is the bridge crane, which is responsible of loading and unloading the fuel rods in the core. Therefore finding all frames where the crane is positioned over the core can represent a useful asset for identifying if the fuel replacement campaign proceeded as declared. For this use-case the SVM was trained on the data of 4 days of 2021 and evaluated for the data of 13 days (5 days of 2021 and 8 days of 2022). In Table 3 we can see the average results after running 9 independent experiments using 128 positive and 128 negative labels.

	Accuracy		Precision		Recall	
	μ	σ	μ	σ	μ	σ
Frames	0.9184	0.0222	0.8597	0.0367	0.9867	0.0140
Events @ TO 90%	0.9675	0.0277	0.5236	0.0210	0.9675	0.0277
Events @ TO 70%	0.9912	0.0175	0.9468	0.0388	0.9912	0.0175
Events @ TO 50%	0.9957	0.0105	0.9891	0.0241	0.9957	0.0105

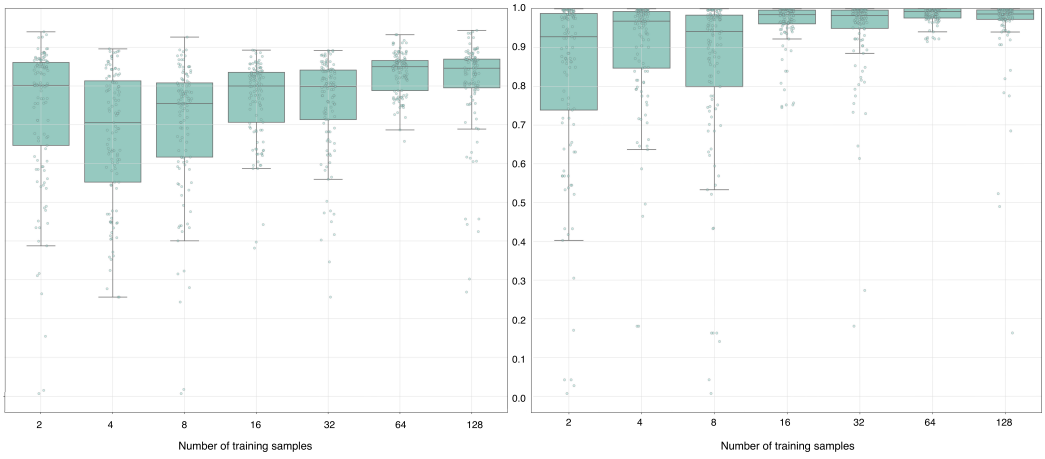
Table 3: Performance results on the bridge dataset.

Our experiments show that the model is able to overall identify relevant events with really high precision and accuracy results. However, the most restrictive scenario (i.e. TO 90%) shows lower precision scores, indicating that the model may be too sensitive in some scenarios, thus predicting them as relevant events. This may be due to the nature of the use-case since the classification into relevant or irrelevant of the frames is subject to the position of the bridge in the linear axis and any inconsistency during the labelling process may affect the final predictions.

3.4 How important are skills during the review?

Interactive deep learning methods heavily rely on user input, which plays a key role in the training of the model. The input from the user should not only be consistent and contain no mislabeled samples but also balanced in representing the different classes. In addition to this, we should consider the variance of the labelled classes, since some classes may contain images that represent the same object but can be significantly different visually, therefore making it necessary to cover all visual differences when labelling. This dependence on user input rises the question of how significant are the user skills during the training. To what extent can the user labels bias the model? In this section we will try to answer to this question, always under the assumption of consistent (no labelling errors) and balanced (same amount of positives and negatives) labels.

We evaluated our method by training it multiple times with different number of samples so we could assess the impact of the user contribution to the final results. Remember that this dataset was created by selecting one label for each temporal slice in order to reduce the existence of duplicate samples. We selected the bridge use-case for this task since we



a) Precision *b) Recall*

Figure 5: Model performance results for frame classification on the bridge use-case.

considered it the most complex. In Figure 5 each box-whiskers plot represents 9 independent experiments, which were held over 13 videos each and with 7 different dataset sizes. The small circles represent the scores for each video.In total, more than 1.1 million frames were evaluated.

We can observe the performance results evaluated with several metrics. The experiments show a positive trend with respect to the number of labelled samples. We can also see a reduction of the variance of the scores as more labels are used for training. The model produces desirable results with 64 or more labelled samples, leading to high accuracy and recall scores. This number represents an upper bound for the labelled dataset size since the results were produced with random subsets of the training data. An experienced user would need fewer samples.

4 Conclusions and Outlook

In this work we introduced an interactive few-shot machine learning pipeline for relevant frame retrieval on video footage that can be tuned online during the review process. It consists of a deep feature extractor for encoding image samples into meaningful embeddings and a SVM for relevant frame classification. In our experiments we evaluated the proposed pipeline on three different real use-cases with positive results and high recall.

In future works we will focus on facing unbalanced or inconsistent datasets to reduce even further the impact of a biased labelling during the inspector's review. In addition, we would like to explore the effects of using other feature extractors or more advanced techniques such as label propagation. We intend to integrate the trained model into the inspector workflow based on NGSR.

References

[1] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, 2017.

[2] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] C. Yu, J. Wang, C. Gao, G. Yu, C. Shen, and N. Sang. Context prior for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[4] E. Apostolidis, G. Balaouras, V. Mezaris, and I. Patras. Combining global and local attention with positional encoding for video summarization. In *2021 IEEE International Symposium on Multimedia (ISM)*, 2021.

[5] W. Zhu, J. Lu, J. Li, and J. Zhou. Dsnet: a flexible detect-to-summarize network for video summarization. *IEEE Transactions on Image Processing*, 2021.

[6] F. Mao, X. Wu, H. Xue, and R. Zhang. Hierarchical video frame sequence representation with deep convolutional graph network. In L. Leal-Taixé and S. Roth, editors, *Computer Vision – ECCV 2018 Workshops*. Springer International Publishing, 2019.

[7] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *Computer Vision–ECCV 2014: 13th European Conference*, 2014.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: a large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[9] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield. Training deep networks with synthetic data: bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[10] E. Wolfart, A. Casado Coscolla, and V. Sequeira. Deep learning for video surveillance review. In *INMM 63rd Annual Meeting*, 2022.

[11] K. Cao, J. Ji, Z. Cao, C.-Y. Chang, and J. C. Niebles. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[12] K. Doshi and Y. Yilmaz. Any-shot sequential anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

[13] M. Chegini, J. Bernard, P. Berger, A. Sourin, K. Andrews, and T. Schreck. Interactive labelling of a multivariate dataset for supervised machine learning using linked visualisations, clustering, and active learning. *Visual Informatics*, 2019.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *International conference on machine learning. PMLR*, 2021.