

Leveraging Zero-shot Prompt Design for Multi-modal Animal-Vehicle Collision Avoidance*

Ashima Garg^{*1}, Dr. Sonali Gupta² and Dr. Payal Gulati³

J.C. Bose University of Science and Technology, Y.M.C.A. Faridabad, Haryana

Abstract

In recent years, technology has rapidly advanced, leading to a growing demand for smarter architecture. Smart cities, born from this progress, have become an essential requirement in today's world. What sets these smart cities apart from traditional ones is their integration of advanced infrastructure and technology. Ensuring the safety of citizens on the road, especially with the rapid development of parallel industries like self-driving cars, is a primary concern. However, there isn't an abundance of data that comprehensively covers every aspect of data distribution present in the real-world environment for cars, such as various weather conditions like "rainy," "sunny," "foggy," etc. Additionally, the process of gathering and subsequently training on this data can be both computationally and financially demanding. In light of the aforementioned challenges, we present an advanced animal classification model using zero-shot learning, leveraging CLIP—a pre-trained multi-modal model trained on 400 million images, with 63 million for the text encoder and 340 million for the image encoder. Our model surpasses the benchmark for zero-shot learning, outperforming even human performance, with an accuracy of 93.5% compared to human performance at 53.7% for zero-shot learning. The model also excels in one-shot and two-shot performance, achieving 75.7%. Furthermore, we assess the model's accuracy on the ImageNet dataset, where it significantly enhances accuracy from 11.5% to 76.2%, even matching the performance of ResNet-50, despite the use of a mere 1.28 million crowd-labeled dataset for training. Finally, we evaluate our dataset on the *STL10* dataset, where our model achieves nearly 100% or more specifically 99.3% accuracy in identifying the animals present in the dataset, despite not being trained on this dataset.

Keywords

Zero-Shot Learning, Prompt-Engineering, Classification, CLIP, Foundation Models

1. Introduction

With the ongoing growth of the human population, the significance of sustainable development becomes increasingly apparent. Achieving sustainable development requires a careful equilibrium between preserving the environment and progressing human activities. Consequently, our research now shifts its focus to tackle the issue of animal identification on roadways within smart cities. Our primary objective is to reduce road accidents resulting from encounters with animals, thus improving overall safety. Recognizing the critical nature of this problem, our investigation has uncovered a shortage of available datasets. Although datasets exist, their suitability for the specific context is lacking, posing a serious risk if systems are built upon


SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07-12-2023, New Delhi, India

*Corresponding author.

✉ ashimagarg80@gmail.com (A. Garg*); sonali.goyal@yahoo.com (Dr. S. Gupta); gulatipayal@yahoo.co.in (Dr. P. Gulati)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

them. To tackle this, we harness the power of pre-trained large multi-modal models using zero-shot learning to fine-tune it on animal classification. Recently, it has been discovered that the pre-trained model consist *emergent qualities* that are able to train on very few-data. Taking it as reference we focus our system to achieve the desired task using zero-shot learning or more specifically *prompting*. These models consists of the ability to efficiently encode both images and text. Therefore, we apply prompting to classify the animals classes proposed using this ability of these pre-trained models. This encoding facilitates the classification of our animal subset within the proposed classes. What sets our system apart is its capacity to effortlessly expand the range of detectable classes. This is attributed to the inherent capability of these pre-trained models to efficiently encode both visual and textual data. We encode different combinations of the prompts consisting of all entities consisting of the following classes: ‘cow’, ‘dog’, ‘goat’, ‘cat’, ‘zebra’, ‘lion’, ‘leopard’, ‘cheetah’, ‘tiger’, ‘bear’, ‘crocodile’, ‘polar bear’, ‘bull’, ‘camel’, ‘cattle’, ‘duck’, ‘elephant’, ‘rhinoceros’, ‘horse’, ‘monkey’, ‘panda’, ‘gorilla’, ‘ground hog’, ‘donkey’, ‘hippopotamus’, ‘ape’, ‘hyena’, ‘jackal’, ‘meerkat’, ‘chimpanzee’, ‘deer’, ‘lamb’, ‘panther’, and ‘pig’. For example we build the text as “*The image contains a <label> on the road.*” and we find the similarity with the encoded image by the image encoder module of the pre-trained large model in terms of logits. We have used different prompts and after scrutinizing them we were able to extract the most appropriate prompt to classify the animals on the road. We used different prompts, and through extensive experimentation we optimized the prompts manually, some of the other appropriate prompts were: “*There is a picture with a cow on the road*”, “*The image shows a cow standing on the road*”, and “*Cow is seen standing in the middle of the road in the picture*”. Consequently, our objective is to determine whether an animal is present in the recorded media from devices, such as a camera mounted on the car’s hood. To achieve this, we leverage the Contrastive Language-Image Pre-Training (CLIP) model, which has been trained on both text and image data. Contrastive learning, an unsupervised representation learning approach, enables the discovery of hidden data representations without the necessity for manual labeling. This implementation involves grouping similar items together and pairing dissimilar items in other combinations. The optimization process in contrastive learning aims to encourage the model to reduce the distance between entities with similar labels and increase the distance between those with differing labels. They have used this technique using text paired with images found across the internet to predict from the 32,768 randomly sampled text snippets, to which the image was actually paired. As for the evaluation, we have evaluated our method optimized via different engineered prompts on ImageNet, Animal-10 Dataset, and COCO datasets. Basically, we can summarize our contribution using the three points:

- We have proposed a novel system for classification of animals on the road to avoid accidents in smart cities. In our knowledge this system is first of its kind to get proposed which harness the power of pre-trained large multi-modals in order to identify the presence of animals on road to avoid accidents.
- The system is optimized and built on the concept of zero-shot learning, i.e. by leveraging the prompt tuning methodology. This alleviates the need to collect a vast amount of data without mitigating the completeness of the proposed model.
- We have proposed a system that is capable of easily adapting to detect new classes in a zero-shot format without any samples but only with the help of prompt engineering.

The paper is structured by initialing the discussing prior research in the field and our optimizations in Section 2. Then, in Section 3, we present our approach, procedures, and methods applied during the study. This section also outlines how we fine-tune the CLIP model for animal detection. Lastly, we provide a comprehensive summary of our discoveries in Section 4.

2. Related Work

In recent years there has been a tremendous progress in the domain of pre-trained large models. This does not come as a surprise due to their outstanding performance on various independent and identically distributed (IID) dataset and out-of-distribution (OOD) datasets. However, due to their vast size with million and billion of parameters, they are not the first choice that comes to mind when low on computational budget. This is due to their high demand for computational resources to get fine-tuned or trained upon. The concept of “*emergent qualities*” has played a major role to make these models accessible to common researchers. The emergent qualities of these pre-trained large models or *Foundation Models*—a term coined by the research community for these models, consist of using these models via zero-shot learning methodologies, like prompt tuning or in-context learning. In-context learning is the process of sending multiple prompts with labels to the model in a sequence format in one-go to adapt the model to a task with the last prompt with no labels, motivating the model to predict the answer based on the prior prompt and labels received. However, the concept of in-context learning is out of the scope of this paper. As a result, we will be focusing more on prompt-learning—it is process of sending incomplete sentences to the model, motivating it to complete the sentence, revealing the answer/prediction in the process. The goal is to give prompts that are similar to the ones PLM saw during training so as to achieve the downstream task with minimal or no training. These prompts are easy to generate requiring design expertise from humans but for complex downstream tasks, the efficiency of generalization is not good. Automated prompting as the name suggests are the ones that are generated by algorithms. As a result, pre-trained large models are able to perform few-shot [1] and zero-shot learning [2] eliminating the need for expensive data collection to fine-tune for downstream tasks. Prompts given to the model can be either manually or automatically generated. Manual Prompting [3] is done by humans generating prompts that can probe the PLM. Automated prompting has been a recent area of research attraction, which can be categorized as hard prompting [4, 5, 3, 6] and soft prompting [7, 8, 9].

The first method involves using explicit prompts or queries in natural language to interact with the language model. The model processes these prompts, comprehends the context, and generates responses accordingly. In contrast, the second approach involves working with the underlying vector representations (embeddings) of words or phrases within the language model’s internal embedding space. In this embedding space, each word or phrase is represented as a high-dimensional vector. Rather than providing explicit prompts in natural language, embeddings are directly manipulated to achieve the desired outcome. Prompt engineering encompasses more than just reordering words; it also encompasses conveying desired styles, aesthetics, layouts, lighting, and textures. Unlike fine-tuning and pre-training, prompt engineering doesn’t have

an impact on model [10, 11] but has a contextual impact on the result being produced. There have been notable works in the field of animal detection. For example, [12] use a convolutional neural network with an extensive 3.2 million dataset promising real-time detection of 48 animal species and using deep neural networks to automatically annotate the images. Ensuring the quality of such a large dataset is a daunting task, moreover, there can be a higher representation of some animal species leading to biases, and training a deep CNN on a dataset of substantial size necessitates significant computational resources, including powerful GPUs and ample memory, resulting in prolonged training times and resource demands. However, the use of such large models and datasets introduces the risk of overfitting, where the model memorizes training data instead of learning meaningful features. This could impede the model’s ability to generalize effectively to new and diverse data, potentially compromising its real-world performance. Careful consideration and mitigation strategies are essential to strike a balance between resource requirements and the risk of overfitting, ensuring the model’s robustness and adaptability for accurate and reliable results.

Mitigating the above problem [13] propose a two-stage network having ResNet-50 as background and self-attention leading to a feature-pyramidal structure. Two datasets are used for training nearly 60,000 samples which are then fed to the model. Although this study offers a potential solution for object detection challenges. However, its increased complexity may lead to higher resource demands during training and inference, potentially escalating costs. [14] and [15] bring the state-of-the-art YOLO detection models to light. This intricacy might also reduce model interpretability, impacting transparency and accountability. Although trained on a substantial dataset, the model’s ability to generalize to diverse scenarios outside its training distribution could be uncertain. Moreover, the additional complexity might compromise real-time performance, hindering applications with low-latency requirements. The effectiveness of the solution heavily relies on dataset quality, and hyperparameter tuning for the two-stage architecture introduces further intricacies. The former study introduces the YOLOv2 architecture with the inclusion of deformable convolutional layers to address the challenge of geometric variations faced by CNNs. Meanwhile, the latter study employs YOLO-Animal, which utilizes YOLOv5 for detection enhancement through the fusion of a weighted Bidirectional Feature Pyramid Network (BiFPN) and an Effective Channel Attention (ECA) module.

While both approaches contribute to improved geometric generalization, they may encounter limitations in detection tasks. The incorporation of deformable convolutional layers in YOLOv2 could introduce computational complexity and require extensive fine-tuning for optimal performance. Similarly, the fusion of BiFPN and ECA in YOLO-Animal might increase model complexity, potentially impacting real-time processing and hardware deployment. Furthermore, both methods may heavily rely on the quality and representativeness of the training data, potentially struggling with novel scenarios not well-covered by the training dataset. This increased complexity may also compromise interpretability, making it challenging to understand the rationale behind detection decisions. Consequently, it is essential to carefully weigh these trade-offs and conduct thorough validation to ensure the practical applicability and reliability of these approaches across a variety of detection settings.

Noting all the previous works and their contribution, we have proposed a system that is capable of zero-shot learning. The method proposed unlike the aforementioned is computationally cheap and easy to deploy. Additionally it does not require one to need more data to fine-tune

the model for the specific problem. We have proved these points using an extensive evaluation on different datasets which we will cover in the Section 3.

3. Implementation Details

This section is devoted to provide the information about the proposed method in detail. Additionally, through this section we hope to provide the information about the experiments for confirming our hypothesis and credibility of our proposed system. Based on the above details and need we have introduced two subsection, where the former called Proposed Method 3.1 is focused on providing the detailed information about the system proposed in this work. Similarly, in the latter Section 3.2 we showcase the results and performance of our system on various datasets.

3.1. Proposed Methods

We leverage the pre-trained zero-shot ability to adapt to novel tasks. Based on it, we have used prompt tuning to adapt the CLIP model on our custom task of classifying animals on roads. CLIP acts as one of the most thoughtful selection from the existing as models as it is trained on 400 million text-image pairs, granting it the ability of zero-shot learning.

Similar to models like GPT [8], which has popularized these emergent qualities. It creates a 512-dimensional image and text vector which are compared using the cosine similarity using the same vector space. The cosine similarity can be defined using the equation 1. It can be defined as the metric of quantifying the similarity between the two vectors. It measures the cosine similarity between the two vectors, which indicates the similarity between these vectors.

$$\text{cosine similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \quad (1)$$

Here A and B denotes the two multi-dimensional vectors, where the numerator produces the similarity of the magnitude and therefore to purely extract the angle we divide the numerator with the magnitude of these vectors. In other words, firstly the CLIP text encoder module encodes the text into rich text embedding, which is analogous to the vector A , defined above. Similarly, the image encoder module of CLIP encodes the image into rich image features embedding with respect to the textual features. The image features embedding can be related to the multi-dimensional vector B defined in equation 1, which is scaled by a temperature \mathcal{T} and normalized into a probability distribution via the softmax activation function. The highest score of the image-text pairs indicates the close proximity between the image and corresponding text pair. Figure 1 briefly describes the process being referred to, the CLIP model harnesses the text and image where the text encoder derives meaningful feature representations that are semantically rich with all the entities present in the module and being referred to in the prompt leveraging meaningful contextual representations a similar process is executed by the image encoder extracting embeddings out of the image. The extracted embeddings are compared using cosine similarity leading to the classification of animal in the picture. The efficient execution of this task holds significant potential for the advancement of smart cities, where the classification of animals on roads could be seamlessly integrated into self-driving cars to enhance road safety

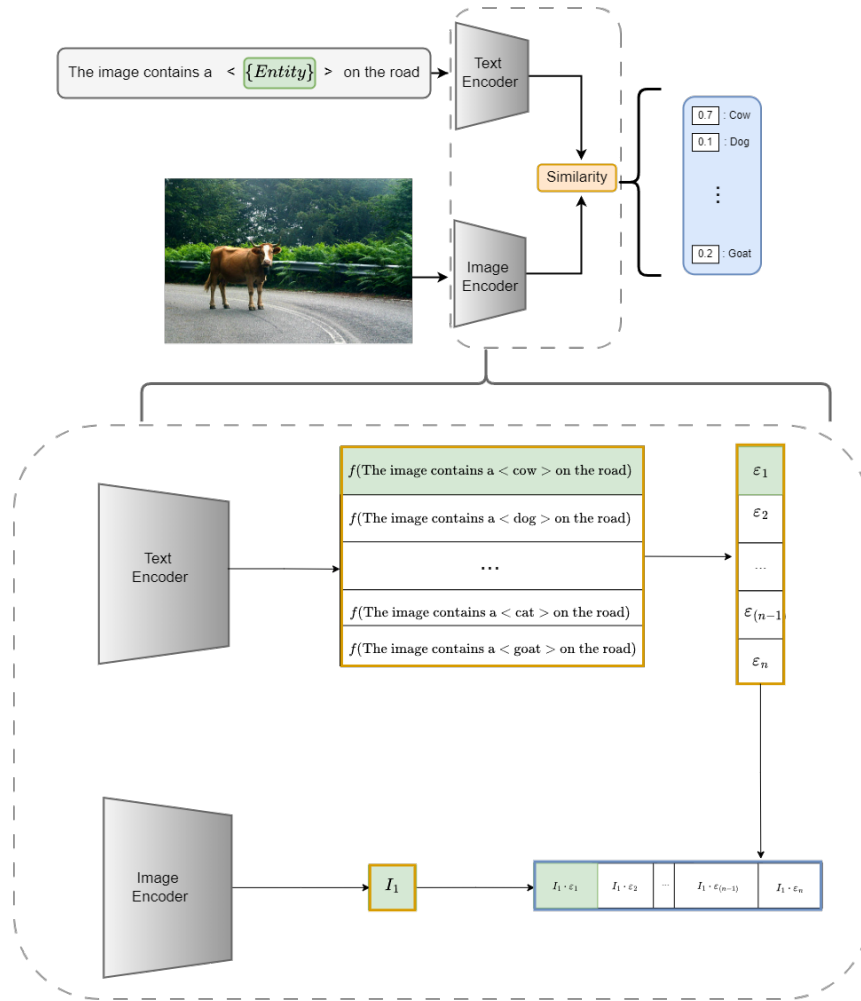


Figure 1: The prompt and the image is leveraged by the CLIP model, where the text encoder is used to obtain the rich semantic feature representation of the text with all the entities available in the $\{\text{Entity}\}$ module. Similarly, the image encoder is used to obtain the embeddings of image. These are both then compared using the cosine similarity to obtain the most appropriate description of image, as a result classifying the animal in the picture.

in India. As previously mentioned, one of the key and distinctive features of our system is its ability to add new classification categories without the need for re-training. While this capability offers substantial benefits, its performance on specific classes must be rigorously evaluated through extensive experimentation.

Prompt Engineering emerged as the most critical and challenging aspect of our system development process. However, it presents a conundrum due to the rapid evolution of Deep Learning, which has made interpretability a substantial challenge. Consequently, crafting the perfect prompt to inspire the model to produce the desired response has become a significant contemporary challenge. Thus, it constituted one of the initial hurdles that we had to overcome.

To overcome it, we leveraged the ChatGPT¹ to build the set of sentences which could be leveraged as prompts for retrieving the classification of the model. We used the query “*paraphrase the sentence given below: a <label> is there on the road*” to get these paraphrased examples from ChatGPT. Specifically, we generated various examples and tested them on our model. The Table 1 describes the top 30 text examples or prompts generated for the model. Furthermore, the table gives a brief overview of the prompts by *ChatGPT* and *Bard* which then served as the prompts guiding the model towards contextual extraction. The variances in prompts by the two models lead to a diversification which in turn enhances the contextual representation ability.

Table 1
Generated Prompts using **ChatGPT** and **Bard** for classification of an animal on the road.

S. No.	Generated Prompt by ChatGPT	Generated Prompt by Bard
1	There’s a <label> present on the road.	A <label> is present on the thoroughfare.
2	A <label> can be observed on the roadway.	A <label> is blocking the pathway.
3	On the road, a <label> is visible.	A <label> is occupying the street.
4	A <label> has positioned itself on the road.	A <label> is obstructing the road.
5	The road features the presence of a <label>.	A <label> animal is on the highway.
6	A <label> has made its way onto the road.	A dairy <label> is on the main road.
7	In the path, you’ll find a <label> on the road.	A <label> is on the asphalt.
8	A <label> is situated on the roadway.	A <label> is on the concrete.
9	The road is home to a <label>.	A <label> is on the blacktop.
10	On the road, one can notice a <label>.	A <label> is on the roadway
11	A <label> occupies space on the road.	There is a <label> on the road.
12	The road hosts the presence of a <label>.	A <label> can be seen on the road.
13	There is a <label> located along the road.	There is a <label> on the road.
14	A <label> is placed on the road.	A <label> can be seen on the road.
15	The roadway accommodates a <label>.	There is a <label> on the road.
16	A <label> is positioned within the road area.	A <label> can be seen on the road.
17	The road has a <label> situated on it.	There is a dairy <label> on the road.
18	Present on the road is a <label>.	A <label> can be seen on the road.
19	A <label> is situated upon the road.	There is a <label> on the road.
20	On the road, there’s the presence of a <label>.	A <label> can be seen on the road.
21	A <label> is positioned in the road’s vicinity.	The <label> is on the road.
22	On the road, a <label> can be found.	The <label> is on the road.
23	A <label> has taken its place on the road.	The <label> is on the road.
24	A <label> is right there on the road.	The <label> is on the road.
25	The road encompasses the presence of a <label>.	The <label> is on the road.
26	In the path, a <label> has appeared on the road.	The dairy <label> is on the road
27	A <label> occupies the space of the road.	The <label> is on the road.
28	The road showcases a <label>’s presence.	The farm <label> is on the road.
29	There’s a <label> positioned on the road.	The livestock <label> is on the road.
30	There’s a <label> positioned on the road.	The <label> is on the road.

¹<https://chat.openai.com/>

3.2. Results and Experimentation

We implemented the proposed technique using Python 3 on the Google Compute Engine backend. At the outset, the code pipeline was constructed with an Nvidia Tesla K80 GPU, endowed with 24 GB of high-speed GDDR5 memory, available at no cost with Colab. Although this GPU served well for executing initial code segments, tasks demanding substantial computational power necessitated careful consideration. The Nvidia Tesla K80 GPU boasts 4992 CUDA cores operating at 560 MHz, translating to training durations spanning approximately 3 to 4 hours. However, training sessions remained confined to 2 to 3 iterations due to sporadic runtime disconnections and GPU memory limitations, rendering the process somewhat intricate and demanding vigilant supervision. To surmount these constraints, we transitioned to Colab Pro, affording us access to the Nvidia Tesla T4 GPU.

We conducted extensive experiments on two distinct datasets, namely ImageNet and STL10. The ImageNet dataset contains a wide range of categories, with the "animal" category alone comprising roughly 3.8 thousand subcategories and 2.8 million images. From this extensive collection, we selected 10 images per unique category, resulting in a total of 38,000 images. Shifting our focus to the STL10 dataset, it encompasses classes like cat, deer, dog, horse, and monkey, each with approximately 800 images.

In addition, we assessed our system's performance in comparison to human abilities. Our model exceeded the benchmark for zero-shot learning, demonstrating superior performance even when compared to human capabilities. Specifically, our model achieved an impressive accuracy of 93.5%, surpassing the human accuracy of 53.7% in zero-shot learning scenarios. Furthermore, our model exhibited commendable performance, achieving an accuracy of 75.7% in both one-shot and two-shot learning scenarios.

Moreover, we evaluated the model's accuracy using the ImageNet dataset, resulting in a significant improvement from an initial accuracy of 11.5% to an impressive 76.2%. Notably, our model's performance aligns with that of ResNet-50, even after utilizing a dataset of 1.28 million crowd-labeled instances for training.

To conclude our assessment, we extended our analysis to the STL10 dataset, where our model achieved nearly flawless accuracy of 99.3% in accurately identifying animals within the dataset. This achievement is particularly noteworthy as our model was not specifically trained on the STL10 dataset.

4. Conclusion

In conclusion, our model presents a compelling stride forward in addressing the challenges posed by diverse environmental conditions, data scarcity, and resource constraints. By harnessing the capabilities of CLIP and zero-shot learning, we contribute a powerful tool for animal classification, not only demonstrating benchmark-beating performance but also showcasing a remarkable ability to generalize beyond its training data. Our proposed model surpasses existing benchmarks for zero-shot learning, outperforming human capabilities with an accuracy of 93.5%, compared to the human score of 53.7% for zero-shot learning. The performance extends to one-shot and two-shot learning scenarios as well, reaching accuracies of 75.7%. Furthermore, our research evaluates the model's prowess on the ImageNet dataset, showcasing

a significant enhancement in accuracy from 11.5% to an impressive 76.2%. Therefore, our model serves as an ideal solution for addressing the challenges of data scarcity and road classification, demonstrating its capacity to detect various environmental conditions encountered in the real world, spanning from "rainy" to "sunny" and "foggy," among others.

However, our model is not without limitations. One prevalent concern is Polysemy, which arises when CLIP's text encoder is provided with only the class name as information. This limitation affects the text encoder's ability to differentiate between different word senses since the absence of context hampers accurate disambiguation.

Furthermore, the introduced model occasionally exhibits limited performance, detecting animals even when they are not present on the road but rather captured by the camera. While this characteristic may be seen as a system limitation, it could potentially serve as a valuable driver warning mechanism to prevent collisions.

References

- [1] W. Bao, L. Chen, H. Huang, Y. Kong, Prompting language-informed distribution for compositional zero-shot learning, 2023. [arXiv:2305.14428](https://arxiv.org/abs/2305.14428).
- [2] R. K. Mahabadi, L. Zettlemoyer, J. Henderson, M. Saeidi, L. Mathias, V. Stoyanov, M. Yazdani, Perfect: Prompt-free and efficient few-shot learning with language models, 2022. [arXiv:2204.01172](https://arxiv.org/abs/2204.01172).
- [3] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How Can We Know What Language Models Know?, *Transactions of the Association for Computational Linguistics* 8 (2020) 423–438. URL: https://doi.org/10.1162/tacl_a_00324. doi:10.1162/tacl_a_00324. [arXiv:https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00324/1923867/tacl_a_00324.pdf](https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00324/1923867/tacl_a_00324.pdf).
- [4] J. Davison, J. Feldman, A. Rush, Commonsense knowledge mining from pretrained models, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 1173–1178. URL: <https://aclanthology.org/D19-1109>. doi:10.18653/v1/D19-1109.
- [5] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, 2021. [arXiv:2012.15723](https://arxiv.org/abs/2012.15723).
- [6] A. Haviv, J. Berant, A. Globerson, BERTese: Learning to speak to BERT, in: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Association for Computational Linguistics, Online, 2021, pp. 3618–3623. URL: <https://aclanthology.org/2021.eacl-main.316>. doi:10.18653/v1/2021.eacl-main.316.
- [7] X. L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, in: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, Online, 2021, pp. 4582–4597. URL: <https://aclanthology.org/2021.acl-long.353>. doi:10.18653/v1/2021.acl-long.353.

- [8] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, J. Tang, Gpt understands, too, 2021. [arXiv:2103.10385](https://arxiv.org/abs/2103.10385).
- [9] G. Qin, J. Eisner, Learning how to ask: Querying LMs with mixtures of soft prompts, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 5203–5212. URL: <https://aclanthology.org/2021.naacl-main.410>. doi:10.18653/v1/2021.naacl-main.410.
- [10] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, 2023. [arXiv:2201.11903](https://arxiv.org/abs/2201.11903).
- [11] S. Garg, D. Tsipras, P. Liang, G. Valiant, What can transformers learn in-context? a case study of simple function classes, 2023. [arXiv:2208.01066](https://arxiv.org/abs/2208.01066).
- [12] B. Nagarajan, S. Srinivasan, Animal detection using deep learning algorithm, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 1–2.
- [13] C. C. Ukwuoma, Z. Qin, S. B. Yussif, M. N. Happy, G. U. Nneji, G. C. Urama, C. D. Ukwuoma, N. B. Darkwa, H. Agobah, Animal species detection and classification framework based on modified multi-scale attention mechanism and feature pyramid network, *Scientific African* 16 (2022) e01151. URL: <https://www.sciencedirect.com/science/article/pii/S2468227622000606>. doi:<https://doi.org/10.1016/j.sciaf.2022.e01151>.
- [14] M. Ibraheam, K. F. Li, F. Gebali, An accurate and fast animal species detection system for embedded devices, *IEEE Access* 11 (2023) 23462–23473. doi:10.1109/ACCESS.2023.3252499.
- [15] D. Ma, J. Yang, Yolo-animal: An efficient wildlife detection network based on improved yolov5, in: 2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), 2022, pp. 464–468. doi:10.1109/ICICML57342.2022.10009855.