**Abstract:**

A novel system is proposed to enable high-level unsupervised and self-supervised learning in a manner that emulates how humans learn from their surroundings. The system is designed to receive visual input using high-resolution cameras, with image preprocessing techniques utilized to create high-quality images and mitigate issues such as illumination variation, motion blur, sensor noise, and atmospheric conditions. Object detection is then performed in real-time using the YOLO software, with object classifications either added to an existing database or entered into an unsupervised learning phase. To improve dataset building, large language models such as GPT-4 are employed to broaden queries and create more accurate object classifications. Backpropagation is used to train the model for each object class, and to prevent unnecessary object re-evaluation, a short-term memory function is implemented. The proposed system differs significantly from traditional object detection and classification algorithms, however without proper experimental results it is hard to determine how effective the system will operate in practice..

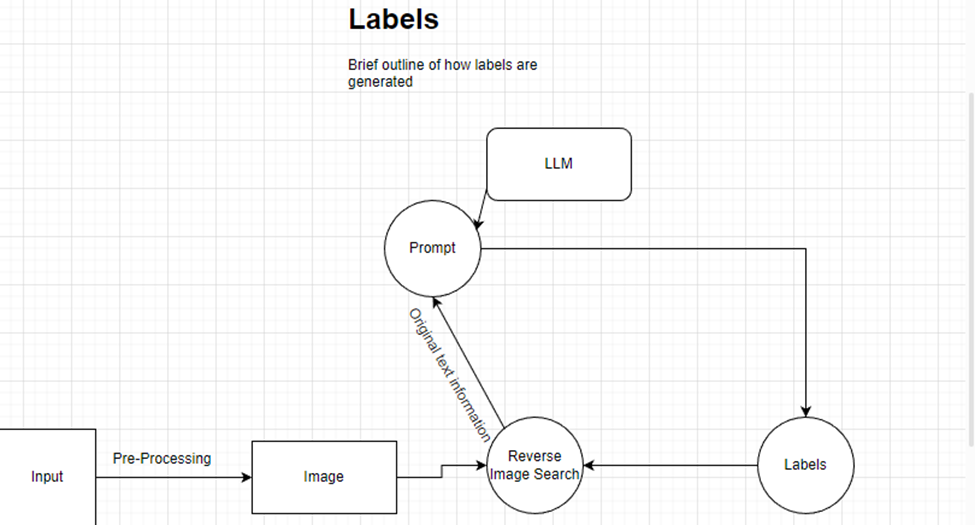
**Introduction:**

Following the release of GPT-4 and its API, humanity seemed to come to a consensus that we have gotten closer to developing Artificial General Intelligence. In fact, some people claim that GPT-4 is already an AGI however this would be considered false, because it is only a highly advanced LLM. For reference, a study from Cornell University titled “Sparks of Artificial General Intelligence: Early experiments with GPT-4. The Microsoft researchers stated “Given the breadth and depth of GPT-4's capabilities, we believe that it could reasonably be viewed as an early (yet still incomplete) version of an artificial general intelligence (AGI) system[1] In Sam Altman’s interview with Lex Friedman, he stated that LLM’s were not the only component created to make a fully functional AGI based on his definition of AGI and the commonly accepted scientific definition of Artificial General Intelligence.[2] Nevertheless, GPT-4 was a step closer to this realization and therefore, we might be able to ask, exactly what things are missing other than LLM’s that are necessary to create an Artificial General Intelligence. The definition of artificial general intelligence is “Artificial general intelligence (AGI) is a hypothetical intelligent agent which can understand or learn any intellectual task that human beings or other animals can.”[3] Based on this definition the widely agreed on “Intelligence traits” The traits necessary for an AGI, are the ability to reason, represent knowledge, plan, learn, communicate in natural language, integrate all these skills for the purpose of achieving a goal, input and output, so the ability to sense the world and manipulate the world.[4] Soon after the release of GPT-4 I released an article on LinkedIn contemplating what would go into creating a Sentient Artificial General Intelligence. By the end writing the short article I had realized that what constitutes sentience and consciousness were both heavily debated so it may be correct to directly define it as sentient, therefore I classified it Pseudo Sentient Artificial General Intelligence. Therefore, this system would pass the Turing test no matter what the level of the evaluator is and be able to greatly imitate all aspects of a human. One of these components is the human optical system. I believe that a high level of unsupervised learning capability is needed to mimic biological systems, therefore there must be an attempt to develop a new system capable of doing so.

Mimicking the biological system of vision requires a combination of hardware and software, the hardware is not very complicated as cameras would have to be used that would be able to mimic the functionality of a biological eye. However human vision is reliant on much more than the eye itself. What me need to focus on is the transcription process of the light waves that hit the retina and transcribing it into images and information. Most existing machine vision software and deep learning software are trained for specific tasks. Existing advanced systems are Google Cloud Vision, Amazon Rekogntition, Microsoft Azure Computer Vision, IBM Watson Visual Recognition, OpenCV, and TensorFlow Object Detection API. All of these software is state of the art Google cloud vision software was trained using a vast amount of data, including labeled images, videos, and text. It uses deep learning models to extract features from images and classify them into different categories, such as objects, text, and facial expressions. Google Cloud Vision also has the ability to detect explicit content and perform optical character recognition (OCR) to extract text from images. [5] Amazon Rekognition uses machine learning algorithms to analyze images and videos. It can detect and recognize objects, people, text, and faces, and can also recognize facial expressions, emotions, and gender. Amazon Rekognition was trained on a large dataset of images and videos, and it uses deep learning models, such as CNNs, to process and analyze the data. (grab citation for this). Microsoft Azure Computer Vision uses CNNs and recurrent neural networks (RNNs), to analyze images and videos. It can detect and recognize objects, text, faces, and emotions, and can also perform OCR and image captioning. Microsoft Azure Computer Vision was trained using a large dataset of labeled images and videos, and it continues to learn and improve over time. (Insert another citation here) IBM Watson Visual Recognition: IBM Watson Visual Recognition uses deep learning models, such as CNNs and long short-term memory (LSTM) networks, to analyze images and videos. It can detect and recognize objects, text, and faces, and can also classify images into different categories, such as food, animals, and landscapes. IBM Watson Visual Recognition was trained on a large dataset of images and videos, and it continues to learn and improve over time. (citation needed) OpenCV: OpenCV is an open-source computer vision library that provides a wide range of functions and tools for image and video processing. It includes algorithms for feature detection and extraction, object detection and recognition, face detection and recognition, and motion analysis, among others. OpenCV was trained using a variety of datasets, including the MNIST handwritten digit database and the CIFAR-10 and CIFAR-100 datasets. (Citation)TensorFlow Object Detection API: This software uses deep learning models, such as CNNs and R-CNNs, to detect and recognize objects in images and videos. It can detect objects in real-time, and it can also be used for object tracking and classification. TensorFlow Object Detection API was trained on a large dataset of labeled images and videos, and it continues to learn and improve over time.(citation) All these models are extremely powerful; however, these models are highly specialized towards certain tasks meaning that they are less versatile and adaptable to new situations. They also lack sufficient unsupervised learning capabilities to be compared to biological systems. The datasets that they are trained on usually contain labels, therefore a large amount of human help is necessary in the learning process of the model. This makes it challenging for the AI system to learn and generalize from unstructured data and limits the ability of the systems to adapt to new situations or tasks without additional labeled data, in contrast biological systems have a remarkable ability to learn from their environment and adapt to new situations without explicit instruction. A new system would be necessary for the purpose of creating AGI. In this paper we explore a potential system that could be used for more general purposes, and functions through a large amount of unsupervised and self-supervised learning.

3. Proposed System

My proposed system is a new system capable of a high level of unsupervised and self-supervised learning in a way similar to how humans learn from their environment. First addressing how the system would receive visual input, The sensory input device does not have much effect on the greater part of the system however, it will be reliant of a relatively high resolution. Therefore, the hardware used for the optical camera for simplicity’s sake is one or two high resolution cameras with a similar field of view to a human being. However, the system can be tested using any high-resolution camera. The camera needs to be able to identify and detect visual noise. The software is multilayered shown in Fig .1. It starts with taking in the visual input through the hardware. We use image preprocessing methods to create a high-quality image, to reduce the risk of unintended noise and artifacts in the image. Sources of distortion are illumination, Occlusions, Motion blur, sensor noise, and atmospheric conditions. Illumination variations are addressed through histogram equalization, adaptive thresholding, our camera also has a high dynamic range. We use the YOLO software which can detect objects even when they are partially obscured. Motion blur can be reduced with a high frame rate, although we are detecting and operating in real time, we still need frames for object classification and detection. Sensor noise can be avoided with high quality materials and hardware, like antireflective coating on the input device. Finally, for the issue of atmospheric conditions effecting the image, we use dehazing techniques, weather-sealed design on the camera, as well as a sensor to detect whether there is an abnormality on the camera like water droplets, dust, or other particles that may affect the image. Now that the system has vision, utilizing the YOLO software we detect separate objects present in the frame in real time. We then take the detected objects and test to see if the object classification exists in the database or if the object is unknown to the system. If the object is known, then we add the sample of the object to the database, and then we adjust the weights of the objects in the database, because we assign higher values to objects that have been encountered in reality. This is a similar concept to anecdotal bias in humans since we assign higher importance to our own personal experiences. If the object is unknown we enter a phase of unsupervised learning. To add a new item to the system we use reverse image searching, however using google images as reference the object queried will most likely be the exact object being observed, this could lead to overspecialization and potential overfitting in the model. Our solution to this is using LLMs (Large Language Models) such at GPT-4 to broaden our query, so that we can build a usable and useful dataset to train our neural networks on. Through this method we can create a dataset of any class of objects. Since many objects do not fit perfectly in one class, datasets will often overlap with each other, giving a more accurate classification of the object. LLM’s can also help in providing and classifying objects with main and sub labels which can be used for dataset building, and for classification.



We train the model for each class using backpropagation. The model will output its predicted result, and we can compare it against an additional reverse imaging search to create the true label of the object. To prevent the system from repeatedly attempting to detect and classify the same object, leading to overweighting of the same object in the system, we use a function similar to short-term memory. We compare the last X frames to the current frame, if the frames are the same then no detection will occur. Given that the frames are different, through YOLO we evaluate if the object itself has changed, since it is not necessary to re-evaluate the entire scene because of the change in one component. This method reduces both processing waste and preserves the quality of the data by preventing unnecessary additional prevalence of the same object on the schema of the system. The prompt that assisting in our reverse imaging must be designed in a way so that we can create a dataset diverse enough to capture the full range of variation in the problem domain, but also specific enough to avoid confusion between two different categories. Given objects x and y, with x being a ceiling fan and y being a floor fan, despite both being fans it would be improper to have a dataset of both ceiling and floor fans, due to the lack of visual similarities between the two objects, this is compounded if we add object Z a hand fan to the mix. Therefore, the best design approximately for the generation of queries is the main category and the most prevalent subcategory of an object.

The way that we decide which neural network to use would also depend on the output label generated by the tandem LLM and reverse image search in order to decide which neural networks will be used. This is better than the other options of simultaneously activating every neural network, as the processing power and multithreading required to perform this process would be very strenuous on most modern systems. It is also unnecessary because most objects are not similar enough to other objects for them to be considered. The other option would be one large neural network, this is also not acceptable for training reasons, and it is also a very inefficient method. By choosing which neural networks are activated based on the label, we can save power and time while maintaining a high level of accuracy. The number of neural networks present in the infrastructure is theoretically finite, however I would like to say that it is infinite, as over time the system will continue to expand itself automatically. If we receive multiple output labels, the LLM will then classify the object based on the most likely context. Natural Language Processing is a great tool for the reason that it is how we express the majority of our thoughts. The prompt would have to be written well enough to give the most likely chance of accuracy, and overriding the output result with human input, is a great tool that helps the system in its training.

Design and Experimentation Process:

(Notice: This is the experimentation design, and no experimentation has occurred)

Objective: To develop an object detection and classification model utilizing YOLO, Inception v3 CNN, and GPT-4 API with python and TensorFlow.

Methodology:

First, after implementation of the object detection method using YOLO, we are able to detect all objects that are present on the screen. We test to see if we can generate a reverse image search query using google images.

Second, we use the GPT-4 API to test which prompt would result in the most accurate and useful labels, we have spot x in the prompt that gets auto filled with the average from the first n search results.

Third, we generate a new query to google images to create the dataset of items, given a relevant label (we find the perfect spot so that our object classes are not too specific but possess enough variation).

Fourth, use Inception v3 CNN, they have guides on how to retrain the network to classify new images, the network is pretrained, however it can be modified to suit my needs and I can call import it through TensorFlow.

Fifth, fine tune the neural network, adjusting the weights using backpropagation, determine if transfer learning using the pretrained CNN is an optimal approach.

Sixth, if test neural networks are > 1 using the same prompt from the original query, generate another label to check if it matches the original label. Our test set consists of real-life objects present in any environment to determine how well the model performs. Based on my prompt engineering tests created a sufficient prompt for combining labels into a classification.

Seventh, Validate system performance through comparison with other models, and through personal review.

Expected Result:

Let’s say one n test objects is a sprite can x we then will capture the frame containing x and run a reverse image search on the bounding box containing x.

(Using an additional model on top of YOLO to extract the branding would probably greatly improve the system).

Through an optimal combination of prompting and averaging the description of the first n images let’s say n =30 images, we will create a label of the object.

We hand this label to the LLM which has a prompt engineered to generate a more general label classification of the object, we then create an image query for object x for a number of reference pictures to build the dataset.

We retrain the dataset using our specific instance of the Inception v3 model suited for the purposes of the task.

Return labels to LLM designating the most probable classification based on the labels outputted, we then print this to the system screen.

Potential use cases:

Robotics: pick-and-place operations would be the easiest and potentially the best use case for the system, with marked locations for each object, if the object is not in the marked place the system identifies the object and moves it to the correct place. Could be used for a robot that does sorting operations.

External Resources:

Conclusion:

The software proposed is for the large part theoretical, as I personally lack the time and resources in order to create the system for experimentation. Many realizations would be made and the architecture would most likely be modified as it was being developed. Although the proposed system can be theoretically run on any device possessing a camera, without a certain level of hardware there may be limitations or inaccuracies because of insufficient quality of hardware.

Advantages:

The proposed system is capable of high levels of supervised and self-supervised learning, this allows for the system to learn from its environment in a similar fashion to how human beings learn from their environment. Since we use reverse image searching the learning process is shorter than in humans, however the system is not as sophisticated as in human beings.

The proposed system is better at general usage purposes than existing systems due to the way that it uses unsupervised learning and interacts with encountered objects to learn how to classify the objects.

The proposed system can be combined with other systems, as well as being used for many purposes such as robotics, and overall is just a larger thought in how we can get closer to artificial general intelligence.

The system can make inferences based on the database of items, reducing dependency on fast and reliable internet overtime, however this, is reliant of sufficient internal data storage, which I don’t believe to be entirely feasible with current technology.

Reverse image searching allows for a majority of objects to be referenceable and gives us the capability to classify a large number of objects using the power of the internet.

Applying different weights to objects in the dataset added through the experience of the system are weighted higher than the weights of the data added through the reverse image searching, this will provide a individual learning experience, as well as applying individual experience despite potential risk of anecdotal evidence bias. I would say that anecdotal evidence bias might be considered a slightly positive effect because a similar thing is observed in humans. However, the weights are good enough so that the effect of anecdotal evidence is not large enough to have too much influence on classification.

Sub

Limitations:

Reverse image searching is a promising approach to construct datasets for unknown objects, but it has its limitations. One of the biggest challenges is the reliance on fast and reliable internet connections to access the required images. However, as the database grows over time and inferences can be made, this dependency can be reduced. We can make inferences based on our database of items. DeepMind’s PathNet has is a framework which would have similar applications.

Another limitation is the processing power to space ratio, as current artificial systems cannot match the level of processing power in biological systems Fig.3 for more details. Kurzweil estimates the computing capacity of the human brain at 20 million billion FLOPS – or 20 (here it comes) quadrillion FLOPS.(citation) Using information from the Intel i9-13900KS, we get a figure of .0326 flops per transistor. Using the theoretical clock speed of the intel processor of and an estimated 25.9 billion transistor count (citation) we arrive at a value of 3 petaflops of processing assuming the space was a perfect sphere, and we could pack it entirely of the space with transistors. 3 petaflops are much less than Kurzweil’s estimate of FLOPS. However, based on Joseph Carlsmith’s estimate the brain delivers 11 petaflops therefore there is not an extremely large difference. The fact is assuming we pack a space the size of a human head with transistors (which is not feasible for wiring and heat reasons). With 10nm architecture and assuming that the transistor is a cube we can theoretically achieve approximately 27% of the brain’s power based on this combination of theoretical and real numbers. When considering the facts that real FLOPS is very often less than theoretical FLOPS, the number is much further from the brain’s power. All of the brain’s power is not utilized in optical processes, however “More than 50 percent of the cortex, the surface of the brain, is devoted to processing visual information,” points out Williams, the William G. Allyn Professor of Medical Optics (citation needed) Which would mean we are still short. We estimate the necessary number of flops through.



Where Bf equals the maximum theoretical FLOPS achievable in our artificial brain, Nt equals the number of transistors (calculated based on the volume of the space and volume of the transistor) Ft is the number of flops per transistor (approximated through FLOPS/n for any processor) and Cs is the clock speed given a certain processor.

When solving for Bf given Ft and Cs we can approximate the amount of space needed to reach the same processing capabilities of the estimates of the human brain in pure theory. Theoretical FLOPS can be substituted for Real FLOPS by removing Cs, giving a more accurate number to the needed space. Regardless, when accounting for the other infrastructure besides just the transistors, we come out to a space bigger than what is ideal to contain the model.

Additionally, storage is also a concern, as it is important to store data in the form of features rather than objects themselves to simulate human memory. However, this may require a very large database depending on the specificity of the stored data relative to the importance of the object to the system. A possible solution would be offloading the computation to a cloud-based service or a high-performance computing cluster for processing. This however compounds on limitation one. In doing so we can also reduce power consumption for the system.

Uncertainty is another limitation, as the full extent of how biological systems function is not yet fully understood, which may result in limitations in the ability to fully replicate them. However, getting as close as possible is still a valuable result.

Reliability and accuracy are also important considerations, as well as the challenge of generalization in human perception. Object recognition occurs on multiple levels and can involve context clues, such as categorizing a plate of food as breakfast, lunch, or dinner based on its contents.

Dependencies are also a concern, as the success of the software is dependent on the use of LLMs in many cases. However, this is a necessary consequence of trying to imitate a complex biological system, as humans use a wide variety of techniques in object recognition. Overall, these limitations highlight the challenges of constructing effective machine vision systems and the need for continued research and innovation to overcome them. GPT-4 API would potentially be the greatest suitor to support this system and to increase its accuracy and object recognition capabilities.

Furthermore, Legality can potentially come into question. The data collection falls within a legal grey area of use, AI systems fall into the grey area in many zones currently. Careful evaluation must be made to make sure that there are no copyright problems when searching for images to train the data on.

An additional limitation would be sensitivity to noise, this system is potentially very sensitive to noise, sources of this noise could come from the hardware, or environmental factors. There are many potential sources of distortion variability or interference. Despite many techniques that can be used to reduce system susceptibility to noise, the system would still be greatly sensitive to noise.

Finally, our last limitation is price; price is always a limitation, especially in the context of the majority of companies and developers. The database required to support the design of this system would easily exceed a billion dollars. The upkeep would also not be cheap either. (Insert Citation here)

References:  
[1]Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., & Zhang, Y. (2023, April 13). *Sparks of artificial general intelligence: Early experiments with GPT-4*. arXiv.org. Retrieved April 16, 2023, from <https://arxiv.org/abs/2303.12712>

[2]YouTube. (2023, March 25). *Sam Altman: Openai CEO on GPT-4, chatgpt, and the future of AI | Lex Fridman Podcast #367*. YouTube. Retrieved April 16, 2023, from <https://www.youtube.com/watch?v=L_Guz73e6fw>

[3] Hodson, Hal (1 March 2019). ["DeepMind and Google: the battle to control artificial intelligence"](https://www.economist.com/1843/2019/03/01/deepmind-and-google-the-battle-to-control-artificial-intelligence). [*1843*](https://en.wikipedia.org/wiki/1843_(magazine)). [Archived](https://web.archive.org/web/20200707223031/https:/www.economist.com/1843/2019/03/01/deepmind-and-google-the-battle-to-control-artificial-intelligence) from the original on 7 July 2020. Retrieved 7 July 2020. AGI stands for Artificial General Intelligence, a hypothetical computer program...”

[4]This list of intelligent traits is based on the topics covered by major AI textbooks, including: Russell & Norvig 2003, Luger & Stubblefield 2004, Poole, Mackworth & Goebel 1998 and Nilsson 1998.

[5] Google. (n.d.). Vision AI &nbsp;|&nbsp; cloud vision API &nbsp;|&nbsp; google cloud. Google. Retrieved April 17, 2023, from <https://cloud.google.com/vision/>

[6]