DECEMBER 2024 | ELEKTORMAGAZINE.COM

elektor

design > share > earn

DIGITAL
BONUS
EDITION

534B
SINCE 1961

AI Guest-Edited Edition

Articles
for Pros, Makers,
and Students!

Raspberry Pi AI Camera

50 AI Key Terms

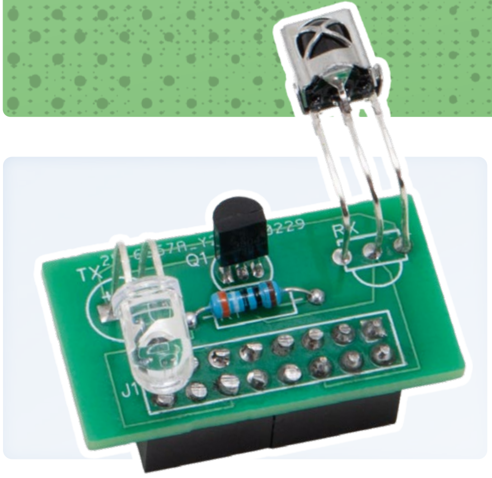
For Engineers
and Makers

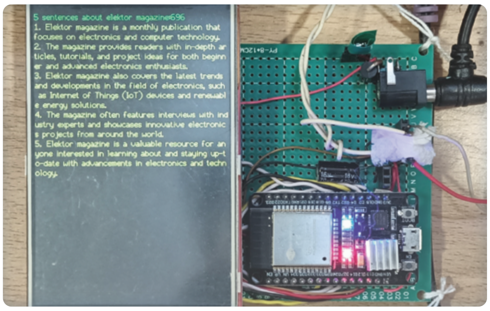
AI for Product Concept Design

A Tour of the World of
AI-Based Art Generators

High-Performance Edge AI

The New STM32N6





```
5 sentences about elektor magazine696
1. Elektor magazine is a monthly publication that
focuses on electronics and computer technology.
2. The magazine provides readers with in-depth ar
ticles, tutorials, and project ideas for both beginn
er and advanced electronics enthusiasts.
3. Elektor magazine also covers the latest trends
and developments in the field of electronics, such
 as Internet of Things (IoT) devices and renewabl
e energy solutions.
4. The magazine often features interviews with ind
ustry experts and showcases innovative electronic
s projects from around the world.
5. Elektor magazine is a valuable resource for an
yone interested in learning about and staying up-t
o-date with advancements in electronics and techn
ology.
```

AI-Based Universal IR Remote Control

Gesture-Controlled with Raspberry Pi

A Personal AI Terminal

ChatGPT Gets a Voice

Hailo AI Kit with Raspberry Pi 5

Object Detection Made Easy

ASUS IoT PE8000G

REVOLUTIONARY EDGE AI PERFORMANCE

Robust and reliable

The PE8000G complies with the MIL-STD-810H standard and is therefore protected against extreme temperatures, vibrations and voltage fluctuations - ideal for use in harsh industrial environments.

Powerful dual GPU support

Process multiple neural networks simultaneously thanks to the support of two 450-watt GPUs - seamlessly and in real time. This allows you to complete even complex tasks efficiently and quickly.

Wide range of applications

From AI-driven factory automation to intelligent video analytics, the PE8000G delivers the computing power and flexibility you need.

Flexible power supply

With a DC input of 8-48V and integrated ignition control, the PE8000G is ideally equipped for a wide range of industrial requirements.

FORTEC
INTEGRATED

Interested in our products?

Volume 50
Guest-Edited AI Digital Bonus Edition
ISSN 0932-5468

Elektor Magazine is published 8 times a year by
Elektor International Media b.v.
PO Box 11, 6114 ZG Susteren, The Netherlands
Phone: +31 46 4389444
www.elektor.com | www.elektormagazine.com

For all your questions
service@elektor.com

Become a Member
www.elektormagazine.com/membership

Advertising & Sponsoring
Büsra Kas
Tel. +49 (0)241 95509178
busra.kas@elektor.com
www.elektormagazine.com/advertising

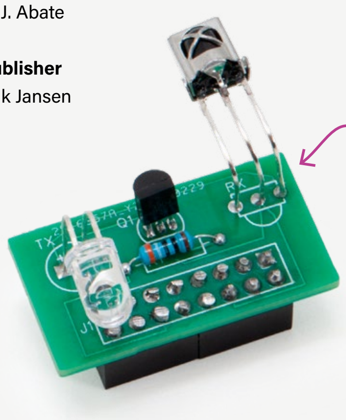
Copyright Notice
© Elektor International Media b.v. 2024

The circuits described in this magazine are for domestic and educational use only. All drawings, photographs, printed circuit board layouts, programmed integrated circuits, digital data carriers, and article texts published in our books and magazines (other than third-party advertisements) are copyright Elektor International Media b.v. and may not be reproduced or transmitted in any form or by any means, including photocopying, scanning and recording, in whole or in part without prior written permission from the Publisher. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature. Patent protection may exist in respect of circuits, devices, components etc. described in this magazine. The Publisher does not accept responsibility for failing to identify such patent(s) or other protection. The Publisher disclaims any responsibility for the safe and proper function of reader-assembled projects based upon or from schematics, descriptions or information published in or in relation with Elektor magazine.

International Editor-in-Chief
Jens Nickel

Content Director
C. J. Abate

Publisher
Erik Jansen



Bonus AI Projects and Articles

Did you really think we were done delivering AI-related content after we announced the 2024 Guest-Edited AI edition of ElektorMag? No way! As regular Elektor readers know, we always have new projects, tutorials, and background articles in the pipeline. With this Bonus edition, which we'll unveil over the course of four weeks, we aim to inspire you to design new AI-related solutions for weeks and months to come.

If you are looking for a project to start experimenting with AI, the article, "AI-Based Universal IR Remote Control," is a great place to start! With the help of a Raspberry Pi 5, Google's MediaPipe Studio platform, and a tiny plug-in interface board, you can have a versatile gesture recognition system to control your TV or other devices operated through IR remote controls.

Can a modern build capture the spirit of Turing's original experiment? In "A Personal AI Terminal," discover how an ESP32 module, paired with a TFT display and an I^2S amplifier, becomes the gateway to a 21st-century version of the Turing Test, communicating directly with ChatGPT.

In "AI for Product Concept Design" we look at how AI has revolutionized image creation. Designers can transform sketches and ideas into photorealistic renderings with just a well-crafted text prompt. Dive into this exploration of AI-powered design tools.

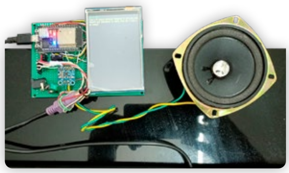
Curious about the history of AI in the pages of Elektor? In this edition, we take you on a journey through Elektor's archives, showcasing how artificial intelligence has evolved within the electronics community. Dive into our editors' top recommendations from past editions of your favorite magazine.

And much more awaits you in this free Bonus edition!

After you dive into this issue and start your own AI-related projects, be sure to share your experiences on the Elektor Labs online platform: *www.elektormagazine.com/labs*. We look forward to learning about your creations!

C. J. Abate (Content Director, Elektor)

THIS EDITION



4 **A Personal AI Terminal**
ChatGPT Gets a Voice

8 **High-Performance Edge AI: The New STM32N6**
A Game-Changer in Performance and AI Capabilities

12 **Unlocking AI**
50 Key Terms for Engineers and Makers

16 **Raspberry Pi AI Camera**
Taking the Hassle out of the Edge

22 **Artificial Intelligence in Elektor**

26 **Hailo AI Kit with Raspberry Pi 5**

32 **Infographics:**
Artificial Intelligence

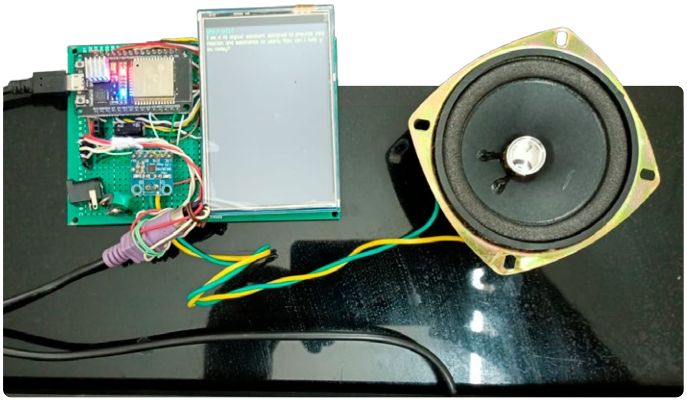
34 **AI-Based IR Remote Control**
Gesture-Controlled with MediaPipe and Raspberry Pi

42 **AI for Product Concept Design**
A Tour of the World of AI-Based Art Generators

The Guest-Edited AI Edition 2024 of ElektorMag is available at newsstands and in the Elektor Store.
www.elektor.com/ep-0534

A Personal AI Terminal

ChatGPT Gets a Voice



By Somnath Bera (India)

Whether AI solutions such as ChatGPT can pass the Turing test is still up for debate. Back in the day, Turing imagined a human operator judging replies to questions sent and received using an electromechanical teletype machine. Here we build a 21st-century version of Turing's original experimental concept, using an ESP32 with a keyboard and TFT display to communicate exclusively with ChatGPT via the Internet. In addition, Google text-to-speech together with a tiny I^{2}S amplifier module and speaker lets you listen in to the conversation. In our case, it's obvious from the start that we are communicating with a machine — isn't it?

There is no doubt that AI tools such as OpenAI's ChatGPT and Google's Gemini can be real game changers in so many situations. I have used ChatGPT to develop quite complex control solutions. I provide the initial idea, and as I give more inputs, it refines the code, making it better with each iteration. It can even convert Python code to MicroPython or an Arduino sketch. The key is to guide the process carefully to prevent it from straying too far off course. There are times when it does deviate and repeats the same mistakes, but I actually enjoy finding these bugs and steering ChatGPT's output to more closely match my target outcome.

Hardware

The heart of this build is Espressif's ESP32 development module. Its dual-core architecture has enough processing reserve to take care of Wi-Fi communications, handling the serial input from a PS2 keyboard, sending data to the 3.5-inch TFT display and outputting digital audio data to the I^{2}S module. The complete schematic for the ChatGPT terminal is shown in **Figure 1**.

You may be surprised at the use of an old PS2 keyboard; the only reason is that I was not successful at implementing a USB port for this purpose on the ESP32. In the limited time available, I opted for the simpler PS2 type of interface, which uses far fewer resources. A PS2 socket for the keyboard can be sourced from several web-based electronic component distributors. The PS2 keyboard pin assignments can be seen in the circuit diagram. No doubt there will be developers out there itching to upgrade the system to USB, and I welcome their input.

The 3.5-inch TFT touch display used has a parallel interface rather than the alternative SPI version. This inevitably uses more interconnect wires, but in this application, we have enough spare GPIOs and the interface offers a significantly faster response time. The touch screen features are not used in this application.

A MAX98357A I^{2}S amplifier (**Figure 2**) module takes the digital audio I^{2}S signal generated by the ESP32 and converts it into analog audio. A built-in, 3-W class D amplifier boosts the signal to achieve a good level of sound. The gain input is tied to ground, which produces maximum volume from the amp. The class D amp output can drive a 4-Ω speaker.

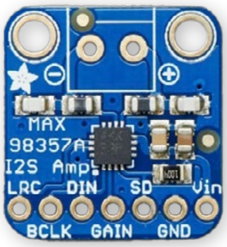


Figure 2: The audio amp module converts I^{2}S digital audio to 3 W of audio.

Don't Lose the Key

ChatGPT was developed by OpenAI. It generates text-based responses to prompts or questions entered by the user. Usually, it works through a browser interface. The OpenAI API has been designed for use by developers and businesses so that they can integrate AI features into their own applications, software, or websites. It's not just a chatbot interface, but a flexible programming interface allowing businesses to embed chatGPT's capabilities into their products. Developers send data to the API programmatically and get responses back. To use the OpenAI API, you will need a unique key, which is generated for your account when you sign up. Go to the OpenAI website [1] and click on the *Sign up* button.

Fill out the registration form with your email address, password, and other required information. Once you have entered these, navigate to your OpenAI account dashboard and click on the *New Project* button. Give your project a name and description (optional). In your project

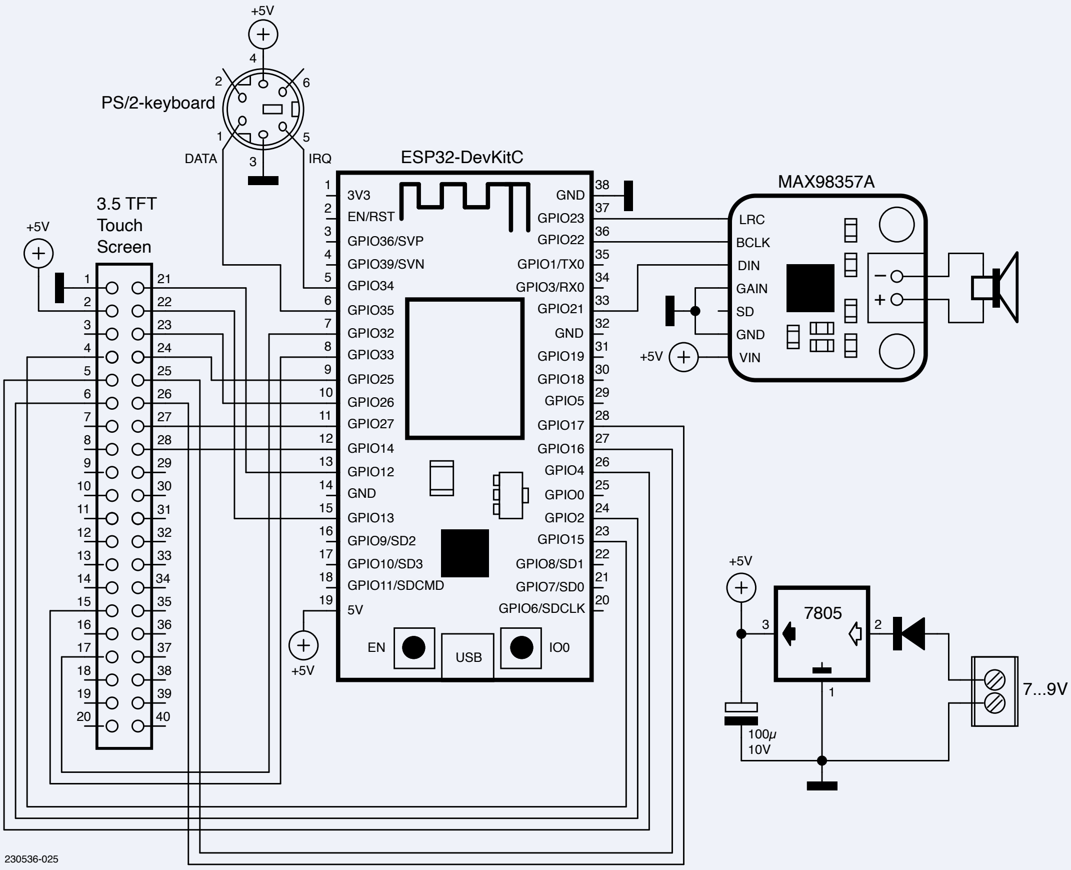


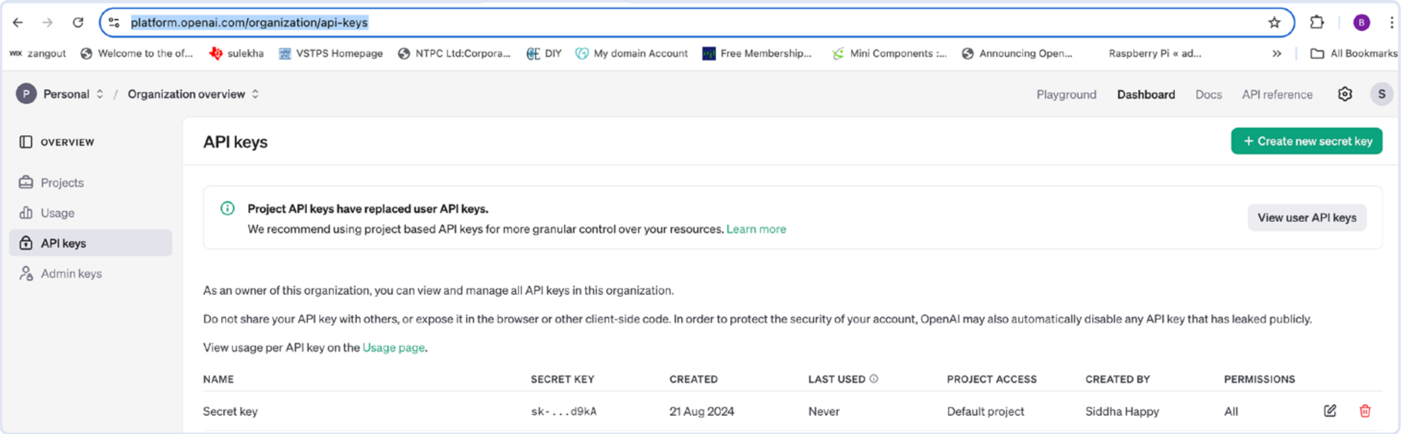
Figure 1: Schematic of the talking ChatGPT terminal.

settings, click on the *API Keys* tab. You'll see a list of existing secret keys (**Figure 3**). Click the *Create new secret key* button to generate a new key. Make a copy of this generated API key and save it securely, as you won't be able to retrieve it again for security reasons. You'll need this key to authenticate your applications with OpenAI's services. On their site there is also a developer's quickstart tutorial which guides you through the process of generating a response via the API using the generated key to authorize your access.

At the time of writing, OpenAI provides free credits when you first sign up, which you can use to experiment with the API. After you use up all these credits, you will need to pay based on your usage. If you haven't already set up a payment method for billing, it will be necessary to do so. Read through OpenAI's usage guidelines and terms of service.

Text to Speech

A Text to Speech (TTS) API is used to convert the textual response from OpenAI into a digital audio data stream. There are a couple of text-to-speech APIs we can use for this. OpenAI has its own TTS API offering a number of different voice alternatives which sound quite natural. To access this API, you use the same key you were allocated to use the OpenAI API.



API keys

Project API keys have replaced user API keys.
We recommend using project based API keys for more granular control over your resources. Learn more

As an owner of this organization, you can view and manage all API keys in this organization.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that has leaked publicly.

View usage per API key on the Usage page.

NAME	SECRET KEY	CREATED	LAST USED	PROJECT ACCESS	CREATED BY	PERMISSIONS
Secret key	sk-...d9kA	21 Aug 2024	Never	Default project	Siddha Happy	All

Figure 3: The OpenAI page to register for the OpenAI API.

For this project, we make use of the Google Cloud Text-to-Speech API. It offers a diverse range of voices in various languages and dialects. Compared to OpenAI TTS the voices sound a little more mechanical. Long text strings make the ouput break up. This API is however free to use at the time of writing, whereas OpenAI TTS API incurs a charge.

To start using the Google TTS, we first need to create the project on Google Cloud and enable the Google TTS API to get our API key. Text strings can now be posted to the API using an HTTP POST request along with the key. The resulting I^{2}S digital audio stream is then stored and played back to produce an analog audio signal via the MAX98357A.

Software: Libraries

The Arduino sketch is included [2]. Check all the libraries referenced in the header of the sketch listing to make sure they are installed in your environment, if not install them now using the Arduino IDE library manager. Using the methods available in the *Audio.h* library, it was a simple job to produce the audio output to accompany the word output written to the TFT display. It was only necessary to add a few lines in the loop to generate the audio. Check it out in the Arduino sketch.

```
#include <PS2Keyboard.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <SPI.h>
#include <TFT_eSPI.h> // Hardware-specific library
#include "Audio.h"    //Audio header file

// GPIO connections to I2S board
#define I2S_DOUT  21
#define I2S_BCLK  22
#define I2S_LRC   23

Audio audio;  //audio instance creation
```

The *ArduinoJSON.h* is used to parse the JSON-formatted response data from the OpenAI API into a format that can be used by the Arduino code.

Software: ChatGPT API request

The process of interacting with the OpenAI API is contained in the `makeApiRequest(String prompt1)` function:

1. First, we set up an `HTTP client`:

```
HTTPClient http;
http.setTimeout(24000); // 24 seconds timeout
http.begin("https://api.openai.com/v1/chat/completions");
http.addHeader("Content-Type", "application/json");
http.addHeader("Authorization", "Bearer " +
                  String(api_key));
```

2. Next, prepare the payload which will be JSON formatted:

```
StaticJsonDocument<1024> jsonPayload;
    // setting a maximum size of 1KB
jsonPayload["model"] = "gpt-3.5-turbo";
    // model is gpt-3.5-turbo,
    // change if you have access to 4 or 4.5
jsonPayload["temperature"] = 0.8;
    // randomness of response, the higher
    // the value, the higher the randomness
jsonPayload["max_tokens"] = 2000;
    // maximum words & punctuations
    // limit to be generated by response
```

3. Prepare the message (a nested array of messages), serialize to JSON format and send the request:

```
JsonArray messages = jsonPayload.
    createNestedArray("messages");
    //put in a nested format not random
JsonObject userMessage = messages.createNestedObject();
userMessage["role"] = "user";
userMessage["content"] = prompt1;
String payloadString;
serializeJson(jsonPayload, payloadString);
```

4. Send the request and receive the response:

```
int httpResponseCode = http.POST(payloadString);
    //transfer to the open page
if (httpResponseCode == HTTP_CODE_OK) {
  String response = http.getString();
    // if everything goes OK get the
    // reply = output and put in a string
...
```

5. The web page will generally include a jungle of spurious information (all in HTML code) which is not useful for our application and can be stripped off. Here we normalize the response — remove the tags JSON etc. so that it can read on the TFT display:

```
StaticJsonDocument<1024> jsonResponse;
    //parse the tags etc of the response string.
deserializeJson(jsonResponse, response);
    // and put in a simple stripped string & return
String assistantReply = jsonResponse
    ["choices"][0]["message"]["content"].
    as<String>();
    // select the first part which contains our reply
return assistantReply;
```

Software: Setup and Loop

In the `setup` function we are initialising the TFT, I^{2}S board, connected to the Internet with our Wi-Fi credentials.

```
void setup() {
  delay(300);
  …
  audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT);
//I2S board initialised
  audio.setVolume(50);
```

Inside the `loop` function, we sent questions to ChatGPT. When the session is complete we remain inside `loop`:

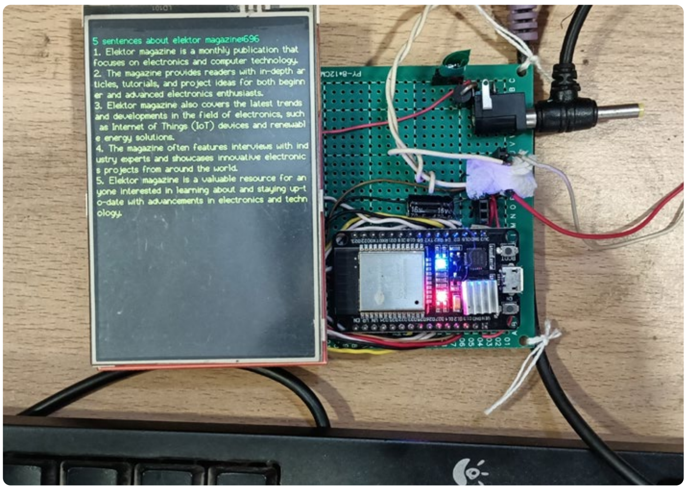
```
String response = makeApiRequest(msg);
    // sent to Chatgpt
...
if (l1>200) {
  response = response.substring(0, 200);
    // truncate first 200 characters
}
audio.connecttospeech(response.c_str(), "en");
    //speak up the 200 characters
```

The Google TTS has a 200-character limit and will not output any text beyond this limit. To work around this, the answer string is trimmed to 200 characters for Google TTS. This ensures that while the full answer is displayed on the screen, only the first 200 characters are vocalized. For longer answers, the screen scrolls, but this can be adjusted by making slight modifications to the sketch.

Project Testing

The delays I've used inside the software loops are quite specific. You may adjust them, but I recommend starting with the default values used in the code. Once you're comfortable with the responses, feel free to tweak them. I began with simple questions like "Who r u?" ChatGPT responded appropriately, displaying the introduction on the screen, while the speaker articulated it clearly.

I then tested the system using questions such as: "Write 5 sentences about Elektor magazine" (**Figure 4**) and even asked it to write blink sketches for Arduino UNO, ESP32, and Raspberry Pi Pico.



```
5 sentences about elektor magazine696
1. Elektor magazine is a monthly publication that
focuses on electronics and computer technology.
2. The magazine provides readers with in-depth ar
ticles, tutorials, and project ideas for both beginn
er and advanced electronics enthusiasts.
3. Elektor magazine also covers the latest trends
and developments in the field of electronics, such
 as Internet of Things (IoT) devices and renewabl
e energy solutions.
4. The magazine often features interviews with ind
ustry experts and showcases innovative electronic
s projects from around the world.
5. Elektor magazine is a valuable resource for an
yone interested in learning about and staying up-t
o-date with advancements in electronics and techn
ology.
```

Figure 4: Testing the system with prompts asking for blink sketches for Arduino UNO, ESP32, and Raspberry Pi Pico.

In every case, ChatGPT performed flawlessly, understood the context perfectly and answered accurately, with the speaker delivering the voice output loud and clear.

To Wrap Up

All files associated with this project can be found at [2]. The Internet and web browsers such as Google have totally revolutionized our access to information and replaced those bulky encyclopedias that used to line our bookshelves at home. Now we see the rise of AI software and machines based on ChatGPT API, TensorFlow Lite Micro, Edge Impulse, OpenMV, TinyML which are poised to disrupt existing business models and more traditional methods of problem-solving. We live in interesting times.

230536-01

About the Author

Somnath Bera, a mechanical engineer from Jalpaiguri Govt. Engg. College, India, worked as a General Manager at NTPC, the largest power producer in the country. He has a profound passion for electronics, evidenced by his 60+ innovative projects on Elektor Labs, over 10 of which have been featured in *ElektorMag*. His projects are often focused on problem-solving in areas like waste and natural resource management. Somnath likes to use innovative approaches and platforms like Arduino, Raspberry Pi, and ESP32 coupled with various kinds of sensors and wireless systems to create efficient and cost-effective solutions.

Questions or Comments?

If you have technical questions or comments about this article, feel free to contact the author by email at berasomnath@gmail.com or the Elektor editorial team at editor@elektor.com.

Related Product

- **Espressif ESP32-DevKitC-32E**
  www.elektor.com/20518

WEB LINKS

[1] OpenAI website: https://platform.openai.com
[2] Elektor web page for this article (downloads):
https://www.elektormagazine.com/230536-01

High-Performance Edge AI: The New STM32N6

A Game-Changer in Performance and AI Capabilities

Contributed by STMicroelectronics

The new STM32N6 is designed to meet the growing demands for high-performance edge AI, enabling new use cases in computer vision, audio processing, and other advanced applications. It continues to elevate the standards for video and multimedia applications, enhancing user experiences with its rich multimedia capabilities.

The STM32 microcontrollers — based on Arm Cortex CPUs from M0 to M7 — are well known in the electronics community, as they are equipped with an abundant set of peripherals and there is a vast ecosystem of development boards, libraries and software tools for all kind of applications. The new STM32N6 addresses the needs for pre-processing data "on the edge", with the help of AI in form of a Neural Processing Unit (NPU). It opens up new fields of applications for Arm Cortex M processors (**Figure 1**).

STM32N6 — Features and Benefits

- The embedded NPU (ST Neural-ART Accelerator) is the heart of the new STM32N6. It offers an impressive 600 GOPS at a very low-power consumption. This NPU is a game-changer, enabling high-end AI performance (**Figure 2**) on an MCU with a small footprint, low power consumption, and low costs.
- Powered by an Arm Cortex®-M55 core operating at 800 MHz, the STM32N6 provides 1280 DMIPS and 3360 CoreMark, marking the highest performance for an STM32 MCU to date. This makes it ideal for advanced vision and graphics applications, supported by a Chrom-ART Accelerator, NeoChrom Accelerator, H264 encoder, and JPEG encoder/decoder.
- The MCU includes 4.2 MB of embedded RAM, supporting real-time data processing, multitasking, and efficient execution without the need for external memory in most cases. Additionally, it offers a rich peripheral set, including dual

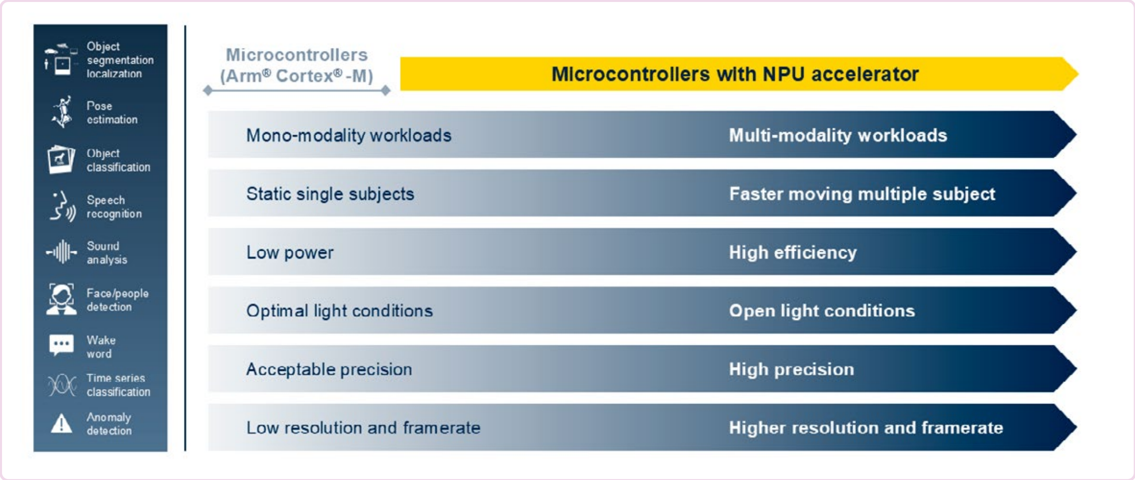


Figure 1: The Neural Processing Unit (NPU) is opening up new fields of applications.

USB, Gbit Ethernet, I3C, and seven UART interfaces, ensuring versatile connectivity options for various applications.

- A dedicated image signal processor (ISP) is dimensioned for a 5 MP camera at 30 fps. Embedded firmware for the Arm Cortex processor allows auto white-balance and auto exposure. A software tool permits fine-tuning of the ISP (**Figure 3**).
- Fully integrated within the STM32 ecosystem, the STM32N6 is supported by the ST Edge AI Suite, including tools like STM32Cube.AI, an AI model zoo, various resources, and case studies. It is also backed by numerous partners, including Edge Impulse, Nota.AI, and EmbedUR, providing a comprehensive development environment.

Comprehensive Software Ecosystem

STMicroelectronics is at the forefront of the edge AI movement, providing comprehensive solutions that cater to the varying needs of engineers and developers across industries. Through its Edge AI Suite, ST offers a wide range of software tools tailored to different user profiles and expertise levels, along with a large collection of case studies to inspire developers.

STM32Cube.AI and ST Edge AI Developer Cloud are designed for users who need great flexibility and customization. These tools provide access to advanced resources for optimizing and deploying custom models (**Figure 4**).

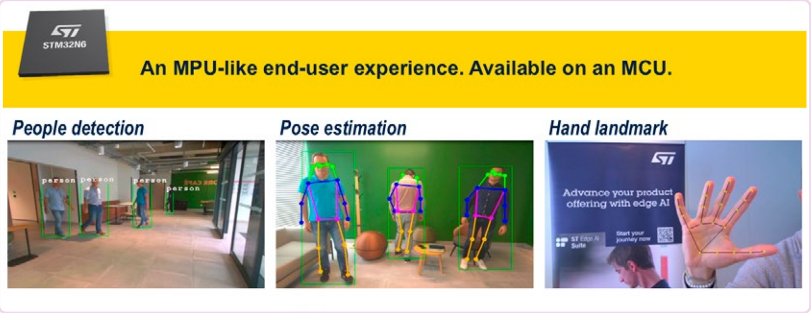

STM32N6
An MPU-like end-user experience. Available on an MCU.
People detection
Pose estimation
Hand landmark
person
Advance your product offering with edge AI

Figure 2: The ST Neural-ART Accelerator enables the execution of advanced AI applications.

Figure 3: A dedicated image signal processor (ISP) is dimensioned for a 5 MP camera at 30 fps and can be fine-tuned via a software tool.

The STM32 Edge AI Contest

Want to use the capabilities of the new STM32N6 in a project of your own? Participate in the **STM32 Edge AI Contest** powered by STMicroelectronics and Elektor! If you are a professional engineer, maker or student, apply for a free STM32N6 development board, and develop your own AI application by the end of April 2025. More information about the contest is available at *www.elektormagazine.com/stm32ai* and in January/February 2025 edition of Elektor.

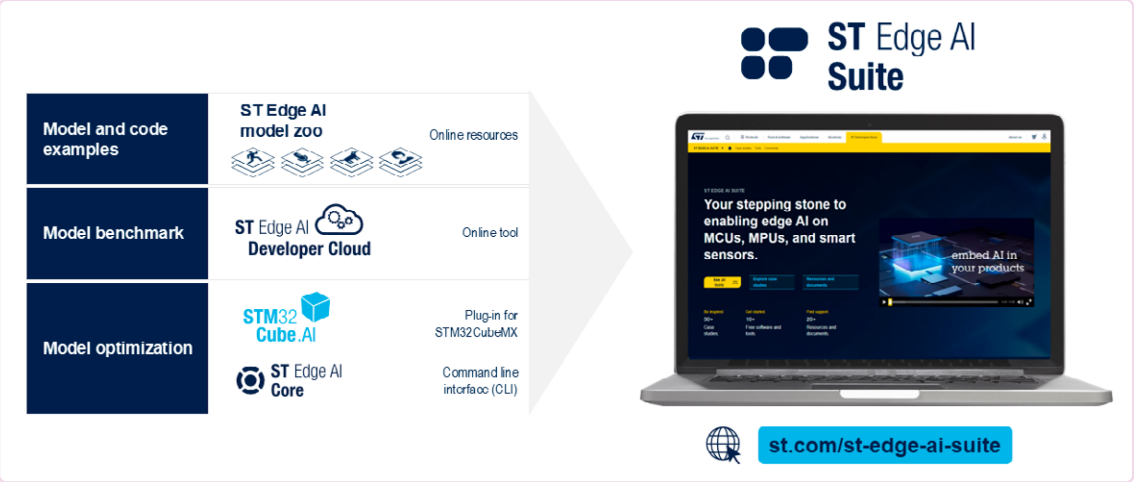


Figure 4: STM32Cube.AI and ST Edge AI Developer Cloud are designed for users who need great flexibility. The comprehensive model zoo with application code examples is a great starting point for developers who want to quickly implement AI solutions.

ST also developed a comprehensive model zoo that comes with additional resources like retraining scripts and application code examples. This is a great starting point for developers who want to quickly implement

People detection

- Application example showing a people detection use case.
- Demonstrating typical AI computer vision application: camera capture, pre-processing, single model inference and post-processing.
- RTOS-based application example.

Multipose estimation

- Application like people detection but built around a multi-pose estimation use case.
- RTOS-based application example.

Hand landmark detection

- Application example showing hand landmark detections.
- Demonstrating the execution of two NN models consecutively.
- RTOS-based application example.

H264 encoding / USB UVC streaming

- Demonstrating a more complete application involving several STM32N6 multimedia features: NPU to perform the inference, H264 encoding and USB video device class stream output data to a PC.
- RTOS-based application example.

Power measurement

- Demonstrating low power optimizations.
- Enabling easy power measurement on STM32N6 discovery board.
- Bare-metal application example.

Figure 5: STM32N6 comes with a set of code examples to kick-start your project.

AI solutions without starting from scratch. The model zoo provides pre-trained models for various applications, such as object detection, image classification, and audio scene recognition, which can be fine-tuned to meet specific requirements. The retraining scripts and application code examples make it easier for developers to adapt these models to their unique use cases, accelerating the development process and reducing time-to-market (**Figure 5**).

It's an STM32!

Developers will benefit from the great ecosystem and legacy of STM32 products to create their own design. The STM32N6 offers a host of advantages for developers, opening up new possibilities and simplifying the development process:

Seamless AI Integration

The STM32N6 is fully integrated into the STM32 ecosystem, making AI deployment straightforward. Developers can leverage the ST Edge AI Suite, including tools like STM32Cube.AI, to optimize and deploy their neural network models with ease. This integration reduces the complexity of AI development and accelerates time-to-market for new products.

Cost and Power Efficiency

By providing MPU-like AI user experience with a smaller footprint, lower power consumption, and reduced costs, the STM32N6 allows developers to create innovative AI-enabled applications without the high expenses typically associated with MPU and GPU solutions.

Enhanced Performance

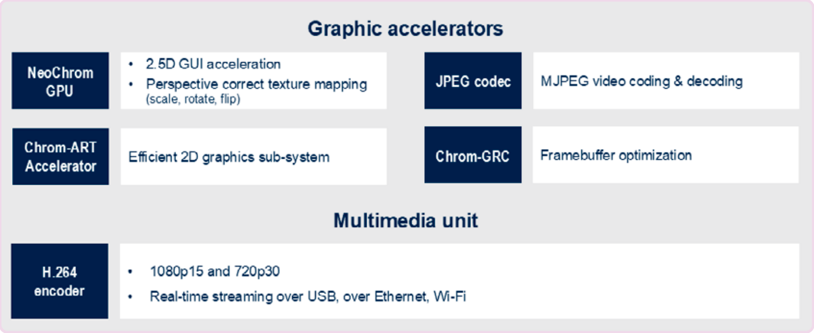
The high-performance Arm Cortex®-M55 core, combined with the embedded NPU and ample RAM, ensures that developers can run complex AI algorithms and multimedia applications efficiently. This opens up new possibilities for creating cutting-edge applications that stand out in the competitive market. The STM32N6's performance capabilities enable developers to push the boundaries of what is possible with MCUs.

Rich Graphics

The STM32N6 stands out as the most powerful STM32 MCU in terms of graphics capabilities. It features the Chrom-ART™ Accelerator for 2D graphics acceleration and the NeoChrom™ Accelerator for 2.5D acceleration, including advanced drawing, perspective correction, and texture mappings (**Figure 6**). Additionally, the Chrom-GRC™ serves as a graphic resource cutter for non-square displays. With 4.2 MB of integrated SRAM, the STM32N6 provides ample memory to handle large graphical assets and complex animations without the need for external memory, ensuring faster and more responsive user interfaces.

Comprehensive Development Environment

The STM32N6 is supported by a robust ecosystem and numerous partners, providing developers with a comprehensive development environment. Leveraging the STM32Cube software suite, it offers



Graphic accelerators

NeoChrom GPU	2.5D GUI acceleration Perspective correct texture mapping (scale, rotate, flip)	JPEG codec	MJPEG video coding & decoding
Chrom-ART Accelerator	Efficient 2D graphics sub-system	Chrom-GRC	Framebuffer optimization

Multimedia unit

H.264 encoder	1080p15 and 720p30 Real-time streaming over USB, over Ethernet, Wi-Fi

Figure 6: The STM32N6 — which is also available without the Neural Processing Unit — is the most powerful STM32 MCU in terms of graphics capabilities.

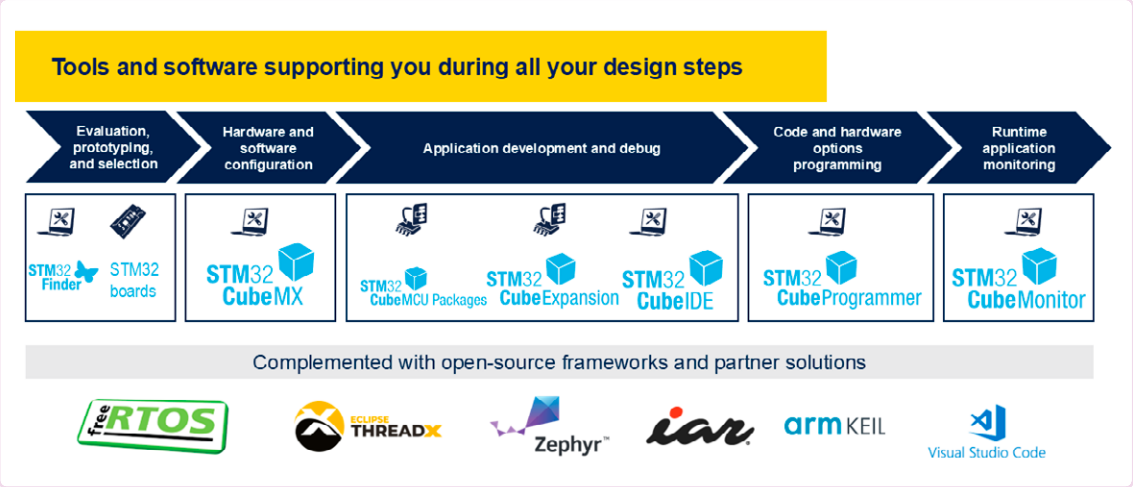

Tools and software supporting you during all your design steps
Evaluation, prototyping, and selection
Hardware and software configuration
Application development and debug
Code and hardware options programming
Runtime application monitoring
STM32 Finder
STM32 boards
STM32 CubeMX
STM32 CubeMCU Packages
STM32 CubeExpansion
STM32 CubeIDE
STM32 CubeProgrammer
STM32 CubeMonitor
Complemented with open-source frameworks and partner solutions
freeRTOS
ECLIPSE THREADX
Zephyr™
iar
arm KEIL
Visual Studio Code

Figure 7: The STM32N6 is supported by a robust ecosystem and numerous partners.

a range of software tools for configuration, development, programming, and monitoring. For embedded software, it includes packages, middleware, and driver support, complemented by RTOS options like FreeRTOS, Zephyr, and Eclipse ThreadX (**Figure 7**).

In the AI domain, developers have access to tools, resources, case studies, and support from partners like Edge Impulse, Nota.AI, and EmbedUR. For graphics development, the STM32N6 is supported by TouchGFX, simplifying the creation of sophisticated graphical user interfaces (GUIs). This extensive support network ensures that developers have the assistance they need at every stage of the development process, enabling the creation of visually stunning and highly responsive applications.

Flexibility and Versatility

With its rich peripheral set and extensive connectivity options, the STM32N6 offers developers the flexibility to design a wide range of applications. Whether it's computer vision, audio processing, or advanced multimedia, the STM32N6 provides the necessary tools and capabilities to bring innovative ideas to life. Additionally, the STM32N6 will also be available in a version without the NPU for applications requiring high-performance computing, graphics, and multimedia but not demanding AI features. This versatility ensures that developers can choose the right configuration for their specific needs, making the STM32N6 suitable for a variety of industries, including automotive, healthcare, consumer electronics, and industrial automation.

Exciting Possibilities for Developers

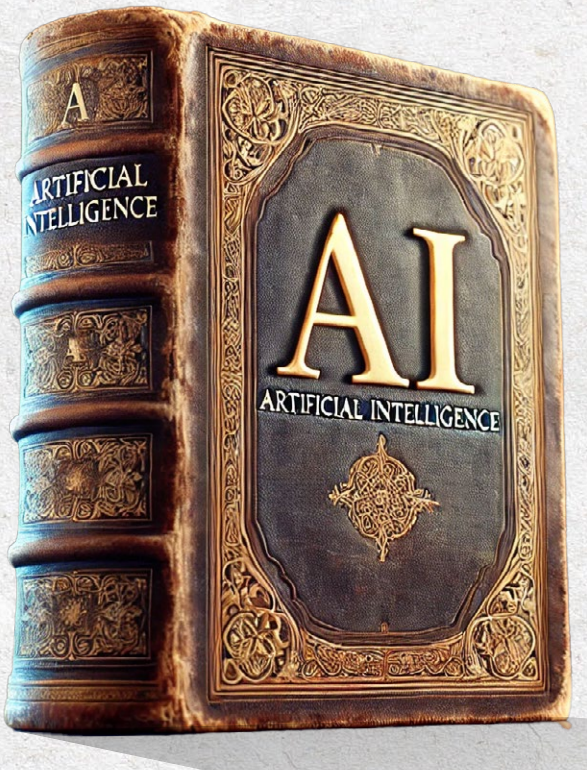
The STM32N6 opens up new horizons for embedded AI applications, offering unmatched edge AI performance on an MCU. Its small footprint, lower power consumption, and reduced costs make it an ideal choice for a wide range of applications, from audio processing to computer vision. The STM32N6 empowers developers to create smarter, more efficient, and more secure devices that can operate at the edge.

Developers can leverage the STM32N6 to create intelligent home automation systems that recognize and respond to various commands and situations, enhancing convenience and security. In healthcare, this STM32 can be used to develop advanced medical devices that monitor patient health in real-time, providing accurate diagnostics and timely alerts. For industrial automation, the STM32N6 enables the design of robust systems that perform complex tasks with high precision and efficiency, improving productivity and safety. In the consumer electronics space, this microcontroller allows for the innovation of new gadgets that offer enhanced multimedia experiences, from high-definition video playback to immersive gaming.

The STM32N6 is a testament to STMicroelectronics' commitment to innovation and excellence. By redefining microcontroller performance and integrating advanced AI capabilities, the STM32N6 empowers developers to create cutting-edge applications that stand out in the competitive market. With its seamless integration into the STM32 ecosystem and support from numerous partners, the STM32N6 is poised to revolutionize the world of microcontrollers and embedded AI.

For more information, visit *www.st.com/stm32n6*.

240676-01



Unlocking AI

50 Key Terms for Engineers and Makers

By The Elektor Team

AI is reshaping the world of electrical engineering and DIY projects. But which terms are essential to know? We asked AI (e.g., ChatGPT and Gemini) — and the answers might surprise you! These terms provide a solid foundation for understanding AI concepts relevant to electrical engineering and making.

Activation Function

A function in neural networks that determines the output of a neuron, helping to introduce nonlinearity. The choice of activation function can significantly impact the network's performance and learning capabilities.

Algorithm

A step-by-step computational procedure for solving a problem or performing a task. Think of it as a guide that outlines the specific steps a computer or person should take to achieve a specific goal.

Artificial General Intelligence (AGI)

Hypothetical AI capable of understanding, learning, and performing any intellectual task like a human. While AGI remains a theoretical concept, researchers are actively working on developing AI systems that exhibit some of these capabilities.

Artificial Intelligence (AI)

The simulation of human intelligence in machines programmed to think, learn, and solve problems. It's a broad field that encompasses a wide range of technologies and applications, including simple algorithms, chatbots, neural networks, and complex self-driving cars.

Backpropagation

A neural network training method that adjusts weights by propagating the error backward from the output layer to minimize the difference between predicted and actual outputs. It updates weights based on the error rate from previous iterations.

Bias

An adjustable parameter in machine learning models that allows for shifting the activation function, influencing model predictions. Bias is an important concept in machine learning, as it can help to improve the accuracy and flexibility of models. However, bias can introduce systematic errors into a model's predictions.

Big Data
Massive datasets that can be analyzed computationally to reveal patterns, trends, and associations. Big Data is commonly used in fields like scientific research and IoT applications to make data-driven decisions, predict outcomes, and optimize processes.

Classification
A supervised learning task of assigning input data to one of several predefined categories or labels. The objective is to build a model that accurately predicts the category of new, unseen data based on patterns learned from labeled training data.

Clustering
An unsupervised learning technique that groups data points with similar features into clusters. Clustering is used for data exploration, pattern recognition, and identifying natural groupings within datasets.

Convolutional Neural Network (CNN)
A deep learning model particularly effective for image and video recognition tasks. CNNs use convolutional layers to automatically learn spatial hierarchies of features from input images, making them well-suited for computer vision applications.

Cross-Validation
A technique for evaluating machine learning models by partitioning data into training and validation sets. It helps assess model performance and generalizability by reducing overfitting and ensuring that the model works well on unseen data.

Data Augmentation
A technique that artificially expands training data by generating variations of existing data samples. Common methods include rotation, scaling, flipping, or adding noise to images, which helps improve model robustness and prevent overfitting.

Data Mining
The process of discovering patterns and extracting useful information from large datasets. It involves methods such as classification, clustering, and association rule learning to transform raw data into actionable insights for decision-making.

Data Preprocessing
The steps taken to clean, transform, and prepare raw data for machine learning models. This includes handling missing values, normalizing data, and encoding categorical variables, which are crucial for ensuring accurate model performance.

Deep Learning
A subset of machine learning using neural networks with many layers to model complex data patterns. Deep learning is particularly effective in handling large datasets with high-dimensional data, such as images, audio, and text.

Dimensionality Reduction
The process of reducing the number of random variables under consideration by obtaining a set of principal variables. This technique helps improve model efficiency, reduce computational costs, and avoid overfitting in machine learning tasks.

Dropout
A regularization technique for neural networks that randomly ignores certain neurons during training to prevent overfitting. By “dropping out” nodes, dropout reduces reliance on specific neurons, improving model generalization on unseen data.

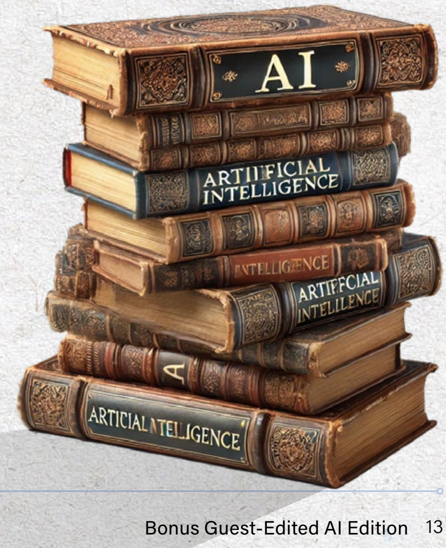
Edge AI
AI computation and processing done locally on edge devices rather than centralized cloud servers. This approach can reduce latency, enhance privacy, and minimize bandwidth use, making it a smart solution for real-time applications like autonomous vehicles and augmented reality.

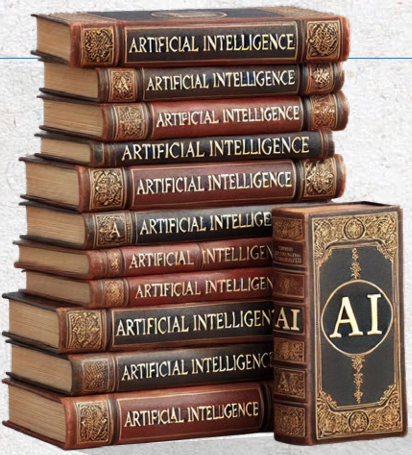
Epoch
One complete cycle through the full training dataset during machine learning model training. Multiple epochs are often required to ensure that the model has learned the data’s underlying patterns adequately.

Feature Engineering
The process of selecting, modifying, or developing new features to improve the performance of machine learning models. Effective feature engineering can significantly enhance model accuracy and reduce overfitting.

Feature Extraction
The process of transforming raw data into a format suitable for machine learning by identifying pertinent characteristics. This step reduces dimensionality and helps in focusing on the most informative attributes for model building.

Generative Adversarial Network (GAN)
A class of machine learning frameworks where two neural networks compete to generate realistic data samples. GANs are often used to create high-quality synthetic images, videos, and even text.





Gradient Descent
An optimization algorithm used to minimize the cost function by adjusting model parameters. It iteratively updates the weights in the direction that reduces the error, helping the model learn from the data.

Hyperparameter Tuning
The process of optimizing the settings of a machine learning model to improve its performance. Hyperparameters (e.g., learning rate or number of layers) are manually adjusted to find the best model configuration.

Inference
The process of using a trained machine learning model to make predictions on new data. Inference is performed after a model has been trained and deployed, and it is often used in real-time decision-making applications.

Internet of Things (IoT)
A network of interconnected devices capable of collecting and exchanging data using embedded sensors and software. IoT enables remote monitoring, automation, and data-driven insights across various industries, from smart homes to industrial systems.

K-Nearest Neighbors (KNN)
A simple, instance-based learning algorithm used for classification and regression by comparing new data points to previously seen examples. It classifies based on the majority class of the nearest data points, making it easy to implement but computationally intensive.

Latent Space
A hidden, low-dimensional space representing compressed features learned by a neural network. Latent space is used in techniques like autoencoders for dimensionality reduction and data generation.

Logistic Regression
A statistical model used for binary classification, predicting the probability of class membership. It uses a logistic function to map input values to a probability, making it effective for distinguishing between two classes.

Loss Function
A mathematical function used to measure the error between predicted and actual outputs in a machine learning model. The loss function guides the optimization process to minimize errors during training, crucial for model accuracy.

Machine Learning (ML)
A subset of AI involving algorithms that learn patterns from data to make predictions or decisions. ML algorithms can be supervised, unsupervised, or reinforced, depending on the nature of the data and the task.

Meta-Learning
A technique in AI where models learn how to learn, adapting quickly to new tasks with minimal data. Meta-learning aims to improve learning efficiency and performance across various tasks by lever0aging previous learning experiences.

Model Compression
Techniques like pruning or quantization that reduce the size and complexity of machine learning models. Compression makes models more efficient for deployment on devices with limited or constrained computational resources.

Natural Language Processing (NLP)
A branch of AI focused on the interaction between computers and human language. NLP enables machines to understand, interpret, and generate human language, facilitating applications like translation, sentiment analysis, and chatbots.

Neural Network
A computational model inspired by the human brain, consisting of interconnected nodes (neurons). Neural networks are the backbone of deep learning, capable of modeling complex, non-linear relationships in data.

One-Hot Encoding
A data preprocessing technique that converts categorical variables into a binary matrix. It enables machine learning models to interpret categorical data by representing each category as a unique vector.

Overfitting
A modeling error in machine learning where the model learns noise instead of the actual pattern, performing poorly on new data. Overfitting occurs when a model is too complex relative to the amount of training data available.

Parameter Optimization
The process of finding the best model parameters to minimize the error during training. This involves adjusting parameters like weights and biases to improve the model's accuracy and performance.

Perceptron
The simplest type of artificial neural network, consisting of a single layer used for binary classification tasks. The perceptron model serves as the building block for more complex neural network architectures.

Recurrent Neural Network (RNN)
A type of neural network that retains information through feedback loops, suitable for sequential data. RNNs are widely used for tasks such as language modeling, speech recognition, and time series prediction.

Regularization
Techniques to prevent overfitting by adding a penalty to model complexity during training. Regularization methods like L1 or L2 norms reduce overfitting and improve model generalization on new data.

Reinforcement Learning (RL)
A type of machine learning where an agent learns to make decisions by receiving rewards or penalties. RL is widely used in robotics, game-playing, and other decision-making scenarios where learning from interaction is essential.

Supervised Learning
A type of machine learning where models learn from labeled datasets to make predictions or decisions. The model is trained using input-output pairs to map inputs to the correct outputs, suitable for tasks like classification and regression.

Support Vector Machine (SVM)
A supervised learning algorithm used for classification and regression tasks, finding the optimal decision boundary. SVM aims to maximize the margin between different classes, ensuring robust classification.

Tensor
A multidimensional array used to represent data in neural networks, especially in deep learning. Tensors provide a framework for manipulating large datasets across multiple dimensions, facilitating efficient computations.

Transfer Learning
A technique where a pre-trained model is fine-tuned on a new, similar task to leverage existing knowledge. Transfer learning is especially useful when training data is scarce or computational resources are limited.

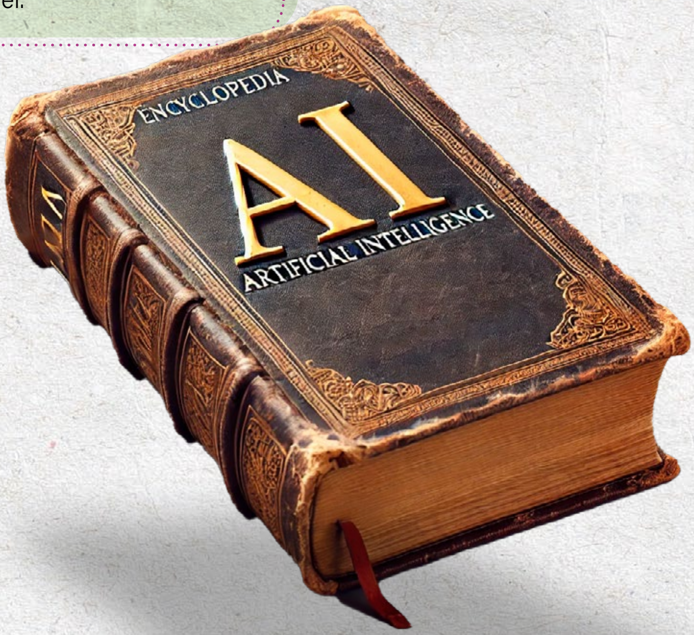
Underfitting
A modeling error where a machine learning model is too simple to capture underlying patterns in the data. Underfitting leads to poor performance on both training and new data, necessitating a more complex model.

Unsupervised Learning
A type of machine learning where models learn patterns from unlabeled data without specific guidance. Unsupervised learning is used for tasks like clustering, dimensionality reduction, and anomaly detection.

Variational Autoencoder (VAE)
A type of neural network used for generating new data samples similar to a given dataset. VAEs learn the distribution of the input data and generate new, similar data points, useful in image synthesis and anomaly detection.

Weights
Parameters in neural networks that are adjusted during training to minimize the loss function. Weights determine the strength of connections between neurons, influencing how input signals are transformed into output predictions.

240535-01



Raspberry Pi AI Camera

Taking the Hassle Out of the Edge

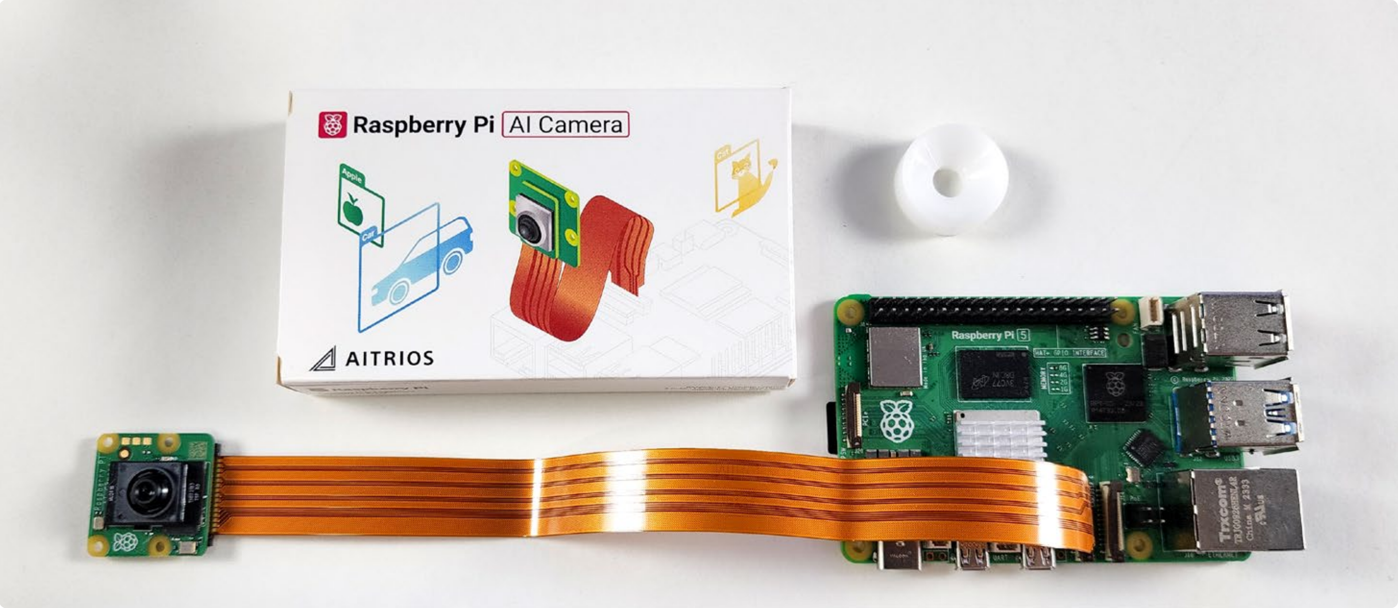


Figure 1: The Raspberry Pi AI Camera (note that the Raspberry Pi is not included).

By Brian Tristam Williams (Elektor)

At Embedded World 2024, I had the chance to see the Raspberry Pi AI Camera. It was unannounced at the time, but it was immediately of interest: a camera with built-in AI capabilities, capable of handling object detection and pose estimation. Fast-forward a few months, and the camera has officially been released. I got my hands on one to put it through its paces.

When my Raspberry Pi AI Camera finally arrived, the box revealed the camera module, two ribbon cables, and — surprisingly — a handy little plastic focus ring (**Figure 1**). In the past, adjusting focus on Raspberry Pi cameras without autofocus meant twisting the lens with your fingers, or worse, using needle-nosed pliers (gently). Not anymore. This tiny, cheap piece of plastic made life easier, even if the lack of autofocus meant you'd still have to fiddle a bit to get it just right.

Under the Tiny Hood

The Sony IMX500 intelligent vision sensor is at the heart of the Raspberry Pi AI Camera, handling both high-resolution image capture and onboard AI processing. Supporting this powerhouse is the RP2040 microcontroller, which manages important peripheral tasks, including firmware management, communication, and ensuring smooth operation between the camera module and the Raspberry Pi.

The RP2040, the same chip used in the Raspberry Pi Pico, is responsible for coordinating communication between the IMX500 and the Raspberry Pi. While it doesn't perform the AI inference, it plays a key role in ensuring that the neural network models and camera functions are loaded and operate correctly. Once the RP2040 sets everything up, the IMX500 handles the AI processing.

By leveraging both the IMX500 for neural network tasks and the RP2040 for control and management, the Raspberry Pi AI Camera efficiently handles AI inference tasks, such as object detection and pose estimation, all without heavily taxing the Raspberry Pi's CPU.

Sony's IMX500 is the world's first intelligent vision sensor with edge processing functionality, meaning it doesn't require a round trip to a server or external accelerator for AI tasks. Instead, all the AI inference is done right on the sensor itself. This design eliminates the need for external processors or high-performance chips, significantly reducing the load on the Raspberry Pi's CPU and allowing for real-time AI-driven applications.

Key Features and Benefits of the IMX500

The IMX500 combines a 12.3-megapixel image sensor with a powerful digital signal processor (DSP) and dedicated on-chip SRAM to perform high-speed AI tasks. This "one-chip solution" stacks the sensor and the AI processing hardware in a single unit, optimizing the entire system for low-latency and low-power AI processing.

Here are some of the standout features of the IMX500:

- 1/2.3" sensor: The IMX500 features a sensor size that supports resolutions up to 4,056 × 3,040 pixels, making it capable of capturing high-quality images while running AI models in real-time.
- Optimized AI processing: The built-in AI accelerator eliminates the need for high-performance external processors or memory, making it a more cost-effective and reliable solution. This translates to lower hardware costs and improved power efficiency.
- Low latency: By handling AI inference directly on the sensor, the IMX500 eliminates the need to send data to a server or external AI accelerator. This significantly reduces latency and ensures that AI decisions are made almost instantaneously.
- Privacy by design: The sensor can output only the processed metadata, ensuring that the raw image data never leaves the chip if privacy is a concern. This feature is particularly useful in applications where privacy regulations are stringent, such as smart retail or smart city deployments.

How the IMX500 Works

The key innovation of the IMX500 is its ability to handle AI inference directly on the sensor. Traditional vision systems typically require image data to be sent to a host processor — whether it's a local CPU, an external AI accelerator, or even a cloud server. This back-and-forth can introduce significant delays, not to mention increased bandwidth and power consumption.

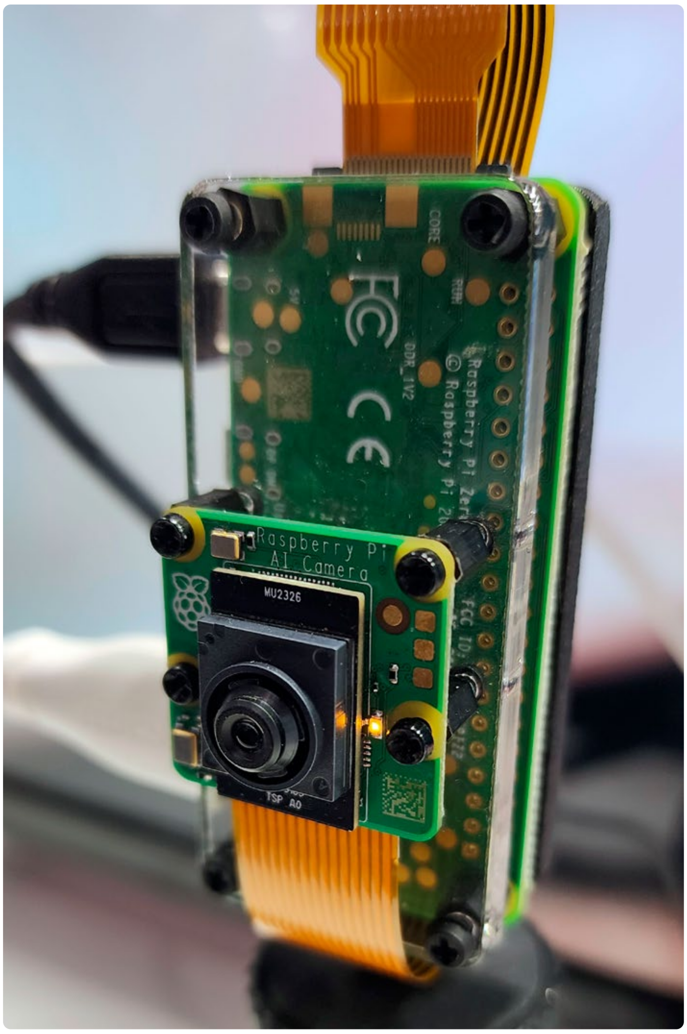


Figure 2: The Raspberry Pi AI Camera won't even tax a Raspberry Pi Zero.

With the IMX500, this processing bottleneck is removed. The sensor itself captures the raw image data, processes it using pre-trained neural networks, and outputs the results as metadata. The Raspberry Pi — any model with a camera connector, from the Raspberry Pi Zero (**Figure 2**) to the Raspberry Pi 5 — receives both the image data and the AI results in parallel, allowing the system to perform tasks such as object detection and pose estimation without any additional hardware or complex processing on the Pi.

The IMX500's ability to crop and scale images using its built-in region of interest (ROI) feature is another great benefit. This allows the sensor to focus on specific areas of the image, ensuring that only the relevant portions are processed by the neural network, which optimizes performance and speeds up AI inference, all without using any of your host system's CPU time.

Sony IMX500 Key Specifications

The key specifications are pretty impressive for a camera that is ostensibly used just for AI applications, which typically don't require super high resolutions:

- Image size: 4,056 × 3,040 — that's a little more than 4K UHD in width, but its 4:3 aspect ratio means 40% more pixels in height.
- Pixel size: 1.55 µm, enabling the camera to capture detailed images even in challenging lighting conditions.
- Frame rate: The camera supports up to 30 frames per second (fps) at a binned resolution of 2,028 × 1,520 pixels and 10 fps at its impressive full resolution, making it suitable for real-time video applications.
- Focal length: 4.74 mm, with a manual adjustable focus ranging from 20 cm to infinity.
- Field of view: Horizontal 66° ±3°, Vertical 52.3° ±3°, providing a wide-angle view suitable for various applications.
- Operating temperature: 0°C to 50°C.
- Size: 25 × 24 × 11.9 mm, this is slightly larger than the Raspberry Pi Camera Module 3, but still compact enough to fit into many existing applications.

AI-Driven Applications with the IMX500

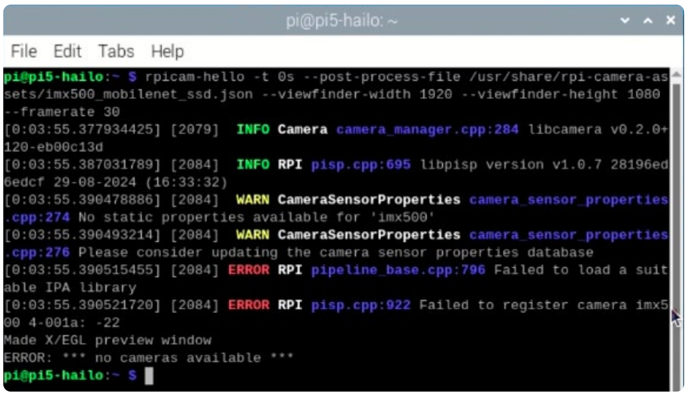
The onboard AI capabilities of the Sony IMX500 open up a wide range of possibilities for vision-based applications. By performing AI inference at the edge, the IMX500 makes it possible to run sophisticated machine learning models without needing additional hardware accelerators such as the Hailo or Coral TPU. Some applications include:

- Object detection: Using the MobileNet SSD neural network, the camera can detect objects in real-time, providing bounding boxes and confidence values for each detected object. This makes it ideal for applications such as security systems, automated retail, or industrial monitoring.
- Pose estimation: With the PoseNet neural network, the camera can track and estimate human poses in real-time. This has applications in healthcare, fitness tracking, and robotics, where understanding human motion is critical.
- Semantic image segmentation: By taking image classification to the pixel level, the IMX500 can classify individual pixels in an image, allowing for advanced use cases such as smart factory monitoring or smart city infrastructure.
- Privacy-preserving vision: Since the IMX500 can process images on the sensor itself and output only the metadata, it's a good fit for applications where privacy is critical. If you so wish, it can analyze foot traffic or detect specific behaviors without ever transmitting raw image data.

Industry Use Cases

The IMX500 has already proven itself in various industries, enabling a wide array of smart vision solutions:

- Smart retail: In retail environments, the IMX500 can be used to analyze customer behavior, detect products, and manage inventory in real-time, all while respecting customer privacy.
- Smart cities: Cities can leverage the camera's ability to classify objects (such as vehicles or pedestrians) and detect dangerous situations in real-time, contributing to safer and more efficient urban environments.
- Smart factories: In industrial settings, the IMX500 can monitor production lines, detect anomalies, and ensure worker safety by tracking movement within designated zones.



```
pi@pi5-hailo:~ $ rpicam-hello -t 0s --post-process-file /usr/share/rpi-camera-as
sets/imx500_mobilenet_ssd.json --viewfinder-width 1920 --viewfinder-height 1080
--framerate 30
[0:03:55.377934425] [2079]  INFO Camera camera_manager.cpp:284 libcamera v0.2.0+
120-eb00c13d
[0:03:55.387031789] [2084]  INFO RPI pisp.cpp:695 libpisp version v1.0.7 28196ed
6edcf 29-08-2024 (16:33:32)
[0:03:55.390478886] [2084]  WARN CameraSensorProperties camera_sensor_properties
.cpp:274 No static properties available for 'imx500'
[0:03:55.390493214] [2084]  WARN CameraSensorProperties camera_sensor_properties
.cpp:276 Please consider updating the camera sensor properties database
[0:03:55.390515455] [2084] ERROR RPI pipeline_base.cpp:796 Failed to load a suit
able IPA library
[0:03:55.390521720] [2084] ERROR RPI pisp.cpp:922 Failed to register camera imx5
00 4-001a: -22
Made X/EGL preview window
ERROR: *** no cameras available ***
pi@pi5-hailo:~ $
```

Figure 3: "No cameras available" persisted at first.

All of this also bodes well for reducing system complexity and cost, while improving reliability and power efficiency.

Setting It Up

So, time for me to get it up and running. As always with me, there were some setup challenges.

Firstly, I the update/upgrade dance as recommended by the official documentation [1]:

```
sudo apt update && sudo apt full-upgrade
```

Then I installed the IMX500 firmware...

```
sudo apt install imx500-all
```

...and did the recommended reboot:

```
sudo reboot
```

That should have been it, and I should have been able to try some object detection using

```
rpicam-hello -t 0s --post-process-file /usr/share/
rpi-camera-assets/imx500_mobilenet_ssd.json --viewfinder-
width 1920 --viewfinder-height 1080 --framerate 30
```

But instead, I got a "no cameras available" error (**Figure 3**). I tried two different cameras, two different cables, and three different Raspberry Pis (two 5s and a 4), and could not get the application running. I checked the forums, did the googling, and all I got were the generic "did you plug it in correctly?" troubleshooting steps. I had to step away, but gave it another go later on. This time, it worked. I wish I could tell you how I solved the problem, but it just came right on its own — just know that if you experience this, you're not alone.

Once the camera was finally recognized, I needed to view the live feed. While I'm not a huge fan of using GUIs on embedded systems, for live camera tasks they certainly have their charms. But still, I didn't want to hook up a mouse, keyboard, and monitor, so I used

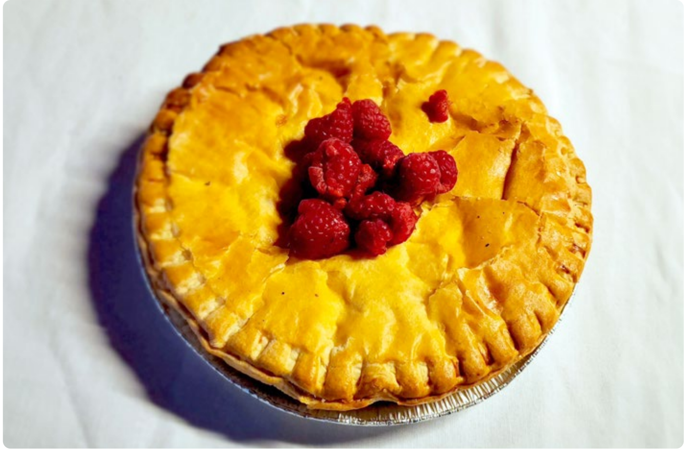


Figure 4: Some raspberries on a pie, or a "doughnut" according to the trained model.

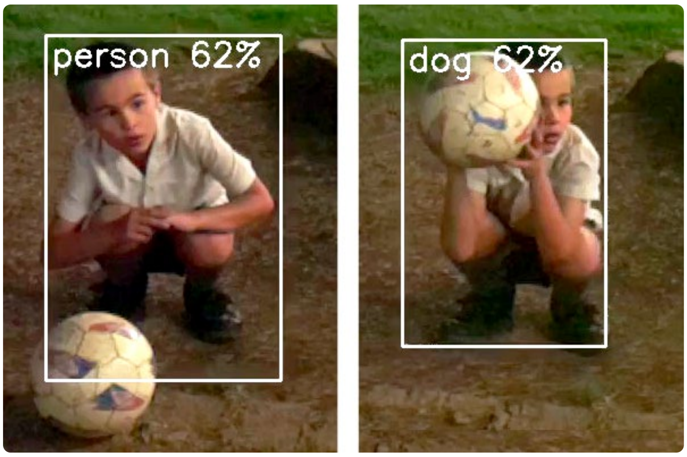


Figure 5: Ashton has the ability to transform between different mammal species.

the still-in-beta Raspberry Pi Connect service [2], since SSH alone wasn't enough for this job. There were a few glitches, but again, it's still in beta.

With the camera working, and Raspberry Pi Connect running on my Windows 11 laptop, I was able to go mobile and try the camera out "in the wild." Since AI cameras will likely be used in various environments, I wanted to test mine outdoors in less controlled lighting conditions.

Object Detection

Using the aforementioned `rpicam-hello` command line, I tried object detection. This identifies objects by providing bounding boxes and confidence values for each object detected. The processing happens directly on the camera, with no significant post-processing needed on the Raspberry Pi itself. Once the output tensor is generated, the system can draw bounding boxes and labels on the image.

I tried a variety of everyday objects, even attempting it to get it to recognize some raspberries. It told me I was likely looking at a "teddy bear." Then I tried a pie, and got nothing. So I put the raspberries on the pie (**Figure 4**), and lo and behold, I had a "doughnut." All fun and games, but the pie was enjoyed, and the children have discovered that they now like raspberries.

Speaking of children, since I wanted some action, I enlisted the help of Chase and Ashton, two tykes who have more energy than I do.

Then I tried some moving targets. Getting Ashton to play with a football yielded a human (**Figure 5**, left), but it wouldn't detect the football, at least under those lighting conditions. I asked him to crouch down and pick up the ball, but this instantly transformed him into a "dog" (**Figure 5**, right).

Pose Estimation

For pose estimation, the camera uses PoseNet, a neural network that labels key points on the body, identifying joints and limbs. Unlike object detection, pose estimation requires additional post-processing on the Raspberry Pi to generate the final output. Once it was running, we were able to overlay stick figures, both in good lighting (**Figure 6**, left), and when it got a bit darker (**Figure 6**, center), but with high-speed tests, such as Chase sprinting past the lens, the camera sometimes lost track of the figures, even at one point chasing Chase (**Figure 6**, right). It also struggled a bit with motion blur, so good lighting would be preferable for reliable results.

Besides taking control of the lighting, you could also tweak the brightness, exposure, and ISO settings from the same command line to maintain decent image quality. Your end application may have to automate this if you can't control the ambient lighting.

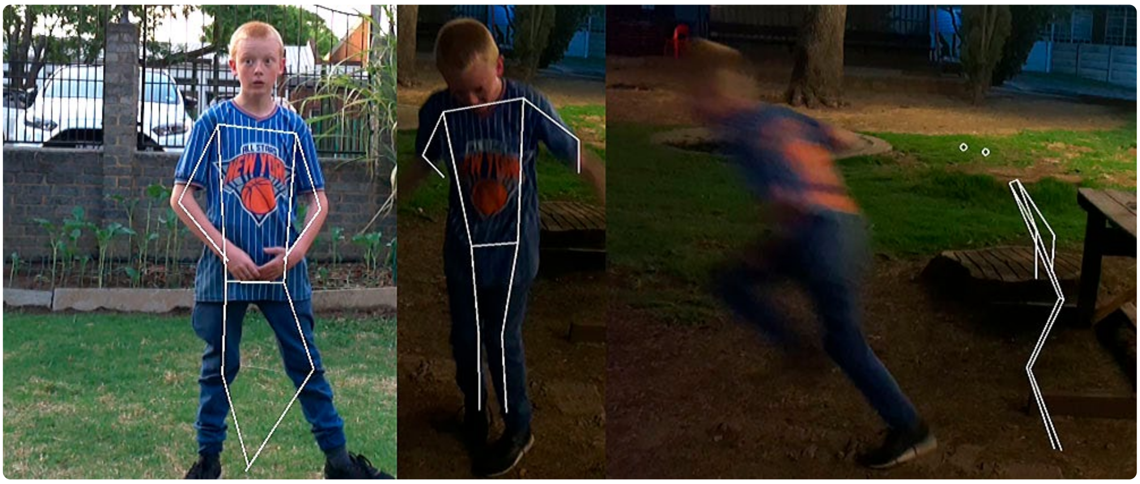


Figure 6: Chase needs to slow down in the dark.

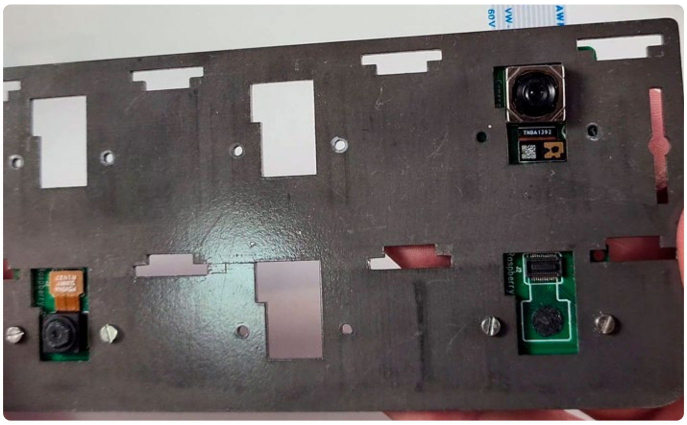


Figure 7: The Raspberry Pi Camera Module 3 and prior models fit into this multi-camera test jig.

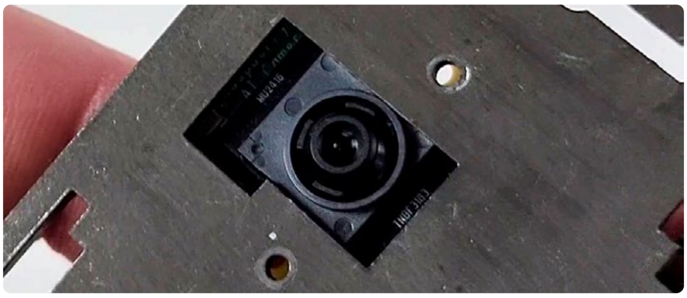


Figure 8: The Raspberry Pi AI Camera may not fit previous camera enclosures.

Mechanical Compatibility

The AI Camera module is slightly larger than older Raspberry Pi cameras, which means it's not 100% compatible with all previous enclosures. I found this out firsthand when I tried to fit the AI camera into a test jig I'd made for comparing multiple versions of the Raspberry Pi Camera Module. While the Camera Module 3 fitted fine (**Figure 7**), the Raspberry Pi AI Camera didn't (**Figure 8**). The third-party market will no doubt fill the gap, but you could always 3D-print something, just as Raspberry Pi themselves did (**Figure 9**) when showing the camera off prior to its official launch.

Lots of Potential

So, now you have an AI Camera that neither taxes your CPU nor requires you to have, for example, a separate accelerator sitting on your Raspberry Pi M.2 HAT+, hogging the bus that you might want to use for an SSD.

Despite the initial challenges, the camera showed real promise. With object detection and pose estimation enabled, I've only scratched the surface of what it can do. The AI processing power of the Sony IMX500 makes this camera a great tool for makers, educators, and anyone wanting to experiment with AI at the edge.

240635-01

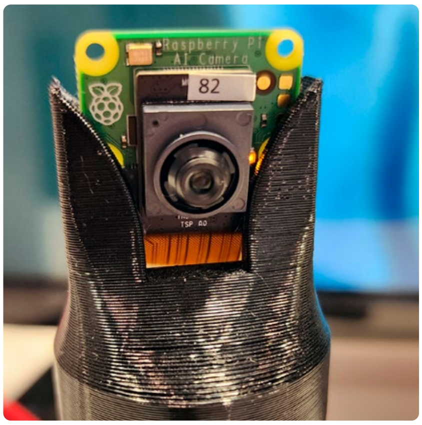


Figure 9: The Raspberry Pi AI Camera was teased with a 3D-printed enclosure.

About the Author

Brian Tristam Williams has been fascinated with computers and electronics since he got his first "microcomputer" at age 10. His journey with Elektor Magazine began when he bought his first issue at 16, and since then, he's been following the world of electronics and computers, constantly exploring and learning. He started working at Elektor in 2010, and nowadays, he's keen on keeping up with the newest trends in tech, particularly focusing on artificial intelligence and single-board computers such as the Raspberry Pi.

Questions or Comments?

We'd love to hear about how you are using AI and how it has affected you over the past two years. If you have questions or comments, email me at brian.williams@elektor.com. You can also catch me on Elektor Engineering Insights each month on YouTube, and you can find me @briantw on X.

Related Products

- **Raspberry Pi AI Camera**
  www.elektor.com/20953
- **Raspberry Pi Zero 2 WH**
  www.elektor.com/20952
- **Raspberry Pi 5 (4 GB RAM)**
  www.elektor.com/20598

FEATURED TOPIC

Visit our **Raspberry Pi page** for articles, projects, news, and videos.

www.elektormagazine.com/
elektor-raspberry-pi

WEB LINKS

[1] Raspberry Pi Documentation: AI Camera: https://raspberrypi.com/documentation/accessories/ai-camera.html
[2] Raspberry Pi Connect remote control: https://www.raspberrypi.com/software/connect

PROTEUS DESIGN SUITE

Driving forward with Manual Routing

Push and Shove Routing for dense layouts

Dedicated Differential Pairs Routing mode

Length Matching and Net Tuning Support

Visual DRC shows legal paths for route placement

labcenter Electronics

www.labcenter.com

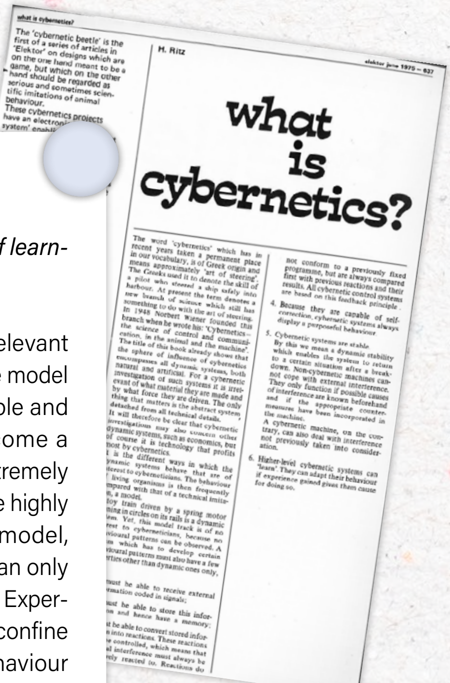
info@labcenter.com

Artificial Intelligence in Elektor

By The Elektor Content Team

Join us on a fascinating journey through Elektor's archives, showcasing how artificial intelligence (AI) has evolved within the electronics community. From early experiments in machine learning to the integration of AI in modern projects, discover how AI has influenced technology trends, inspired innovation, and shaped the future of electronics design. Check out these editor's picks.

Once science fiction, artificial intelligence (AI) is now a cornerstone of modern technology. Elektor, a pioneer in electronics with articles dating back to 1961, has been at the forefront of this evolution, exploring AI's potential since its nascent stages. Whether you manually flip through old editions of *ElektorMag* or use the beta version of our Elektor GPT [1], you will see that AI-related topics have been covered for decades. From discussions of neural networks and fuzzy logic in the 1970s and 1980s to in-depth projects featuring today's sophisticated AI-driven solutions, our coverage has mirrored the swift advancement of AI tech. Here, our editors and engineers offer a selection of significant AI-related articles from our vast archives. Of course, not every article is ostensibly about AI. But the concepts and ideas explored, such as pattern recognition, automation, and machine learning, have been integral to the development of AI tech.



what is cybernetics?

The 'cybernetic beetle' is the first of a series of articles in 'Elektor' on designs which are on the one hand meant to be a game, but which on the other hand should be regarded as serious and sometimes scientific imitations of animal behaviour. These cybernetics projects have an electron system' enabl

H. Ritz

elektor june 1975 — 637

what is cybernetics?

The word 'cybernetics' which has in recent years taken a permanent place in our vocabulary, is of Greek origin and means approximately 'art of steering'. The Greeks used it to denote the skill of a pilot who steered a ship safely into harbour. At present the term denotes a new branch of science which still has something to do with the art of steering. In 1948 Norbert Wiener founded this branch when he wrote his: 'Cybernetics – the science of control and communication, in the animal and the machine'. The title of this book already shows that the sphere of influence of cybernetics encompasses all dynamic systems, both natural and artificial. For a cybernetic investigation of such systems it is irrelevant of what material they are made and by what force they are driven. The only thing that matters is the abstract system detached from all technical details. It will therefore be clear that cybernetic investigations may also concern other dynamic systems, such as economics, but of course it is technology that profits most by cybernetics.

t is the different ways in which the ynamic systems behave that are of terest to cyberneticians. The behaviour living organisms is then frequently mpared with that of a technical imita- n, a model.

toy train driven by a spring motor ning in circles on its rails is a dynamic tem. Yet, this model track is of no rest to cyberneticians, because no vioural patterns can be observed. A m which has to develop certain vioural patterns must also have a few rties other than dynamic ones only.

ust be able to receive external rmation coded in signals;

ust be able to store this infor- on and hence have a memory;

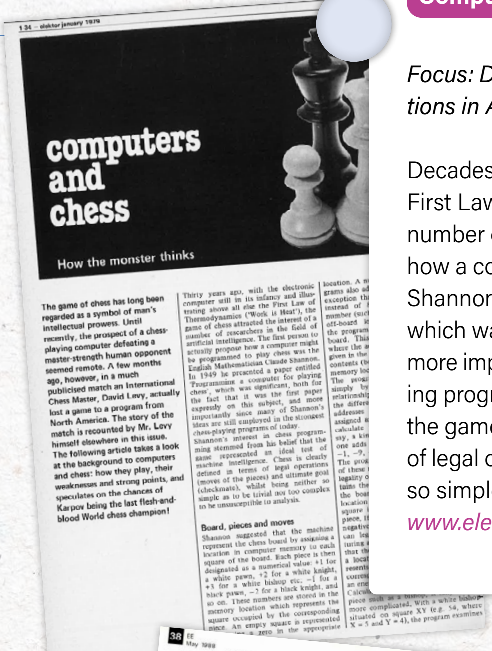
t be able to convert stored infor- n into reactions. These reactions e controlled, which means that l interference must always be ely reacted to. Reactions do not conform to a previously fixed programme, but are always compared first with previous reactions and their results. All cybernetic control systems are based on this feedback principle.

4. Because they are capable of self-correction, cybernetic systems always display a purposeful behaviour.
5. Cybernetic systems are stable. By this we mean a dynamic stability which enables the system to return to a certain situation after a breakdown. Non-cybernetic machines cannot cope with external interference. They only function if possible causes of interference are known beforehand and if the appropriate countermeasures have been incorporated in the machine. A cybernetic machine, on the contrary, can also deal with interference not previously taken into consideration.
6. Higher-level cybernetic systems can 'learn'. They can adapt their behaviour if experience gained gives them cause for doing so.

What Is Cybernetics? (Elektor, June 1975)

Focus: Discusses cybernetic models simulating animal behavior, exploring concepts of learning, adaptability, and intelligence in artificial systems.

The 'cybernetic beetle' is the first of a series of articles in 'Elektor' on designs which are on the one hand meant to be a game, but which on the other hand should be regarded as serious and sometimes scientific imitations of animal behaviour. These cybernetics projects have an electronic 'nerve system' enabling them to react specifically to external acoustic, optical and tactile stimuli. Such models can moreover be designed so that the reaction is not only governed by the type of stimulus, but also by the condition of the model at the moment when the stimulus is applied. The more 'natural' the relationship between the stimulus and the relevant reaction, the more the behaviour of the model approaches that of the animal example and the more will an ingenious toy become a scientific object. It was found to be extremely difficult to imitate the behaviour of more highly developed animals in a cybernetic model, because an electronic nerve system can only perform a limited number of functions. Experiments on a scientific basis therefore confine themselves to certain aspects of behaviour or to living models of a very simple nature. Continue reading: *www.elektormagazine.com/magazine/elektor-197506/57395*



1 34 – elektor january 1979

computers and chess

How the monster thinks

The game of chess has long been regarded as a symbol of man's intellectual prowess. Until recently, the prospect of a chess-playing computer defeating a master-strength human opponent seemed remote. A few months ago, however, in a much publicised match an International Chess Master, David Levy, actually lost a game to a program from North America. The story of the match is recounted by Mr. Levy himself elsewhere in this issue. The following article takes a look at the background to computers and chess: how they play, their weaknesses and strong points, and speculates on the chances of Karpov being the last flesh-and-blood World chess champion!

Thirty years ago, with the electronic computer still in its infancy and illustrating above all else the First Law of Thermodynamics ('Work is Heat'), the game of chess attracted the interest of a number of researchers in the field of artificial intelligence. The first person to actually propose how a computer might be programmed to play chess was the English Mathematician Claude Shannon. In 1949 he presented a paper entitled 'Programming a computer for playing chess', which was significant, both for the fact that it was the first paper expressly on this subject, and more importantly since many of Shannon's ideas are still employed in the strongest chess-playing programs of today. Shannon's interest in chess programming stemmed from his belief that the game represented an ideal test of machine intelligence. Chess is clearly defined in terms of legal operations (moves of the pieces) and ultimate goal (checkmate), whilst being neither so simple as to be trivial nor too complex to be unsusceptible to analysis.

Board, pieces and moves

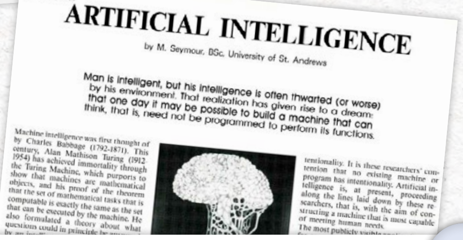
Shannon suggested that the machine represent the chess board by assigning a location in computer memory to each square of the board. Each piece is then designated as a numerical value: +1 for a white pawn, +2 for a white knight, +3 for a white bishop etc. −1 for a black pawn, −2 for a black knight, and so on. These numbers are stored in the memory location which represents the square occupied by the corresponding piece. An empty square is represented

more complicated. With a white bishop situated on square XY (e.g. 54, where X = 5 and Y = 4), the program examines

Computers and Chess: How the Monster Thinks (Elektor, January 1979)

Focus: Discusses the development of chess-playing computers and highlights key contributions in AI, particularly by Claude Shannon.

Decades ago, with the electronic computer still in its infancy and illustrating above all else the First Law of Thermodynamics ('Work is Heat'), the game of chess attracted the interest of a number of researchers in the field of artificial intelligence. The first person to actually propose how a computer might be programmed to play chess was the English Mathematician Claude Shannon. In 1949 he presented a paper entitled 'Programming a computer for playing chess,' which was significant, both for the fact that it was the first paper expressly on this subject, and more importantly since many of Shannon's ideas are still employed in the strongest chess-playing programs of today. Shannon's interest in chess programming stemmed from his belief that the game represented an ideal test of machine intelligence. Chess is clearly defined in terms of legal operations (moves of the pieces) and ultimate goal (checkmate), whilst being neither so simple as to be trivial nor too complex to be unsusceptible to analysis. Continue reading: *www.elektormagazine.com/magazine/elektor-197901/58282*



38 EE May 1988

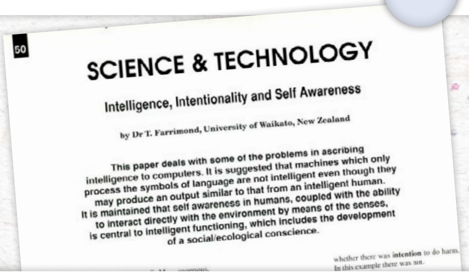
ARTIFICIAL INTELLIGENCE

by M. Seymour, BSc. University of St. Andrews

Man is intelligent, but his intelligence is often thwarted (or worse) by his environment. That realization has given rise to a dream: that one day it may be possible to build a machine that can think, that is, need not be programmed to perform its functions.

Machine intelligence was first thought of by Charles Babbage (1792-1871). This century, Alan Mathison Turing (1912-1954) has achieved immortality through the Turing Machine, which purports to show that machines are mathematical objects, and his proof of the theorem that the set of mathematical tasks that is computable is exactly the same as the set that can be executed by the machine. He also formulated a theory about what questions could in principle be

tentionality. It is these researchers' contention that no existing machine or program has intentionality. Artificial intelligence is, at present, proceeding along the lines laid down by these researchers, that is, with the aim of constructing a machine that is most capable of meeting human needs.



50

SCIENCE & TECHNOLOGY

Intelligence, Intentionality and Self Awareness

by Dr T. Farrimond, University of Waikato, New Zealand

This paper deals with some of the problems in ascribing intelligence to computers. It is suggested that machines which only process the symbols of language are not intelligent even though they may produce an output similar to that from an intelligent human. It is maintained that self awareness in humans, coupled with the ability to interact directly with the environment by means of the senses, is central to intelligent functioning, which includes the development of a social/ecological conscience.

whether there was **intention** to do harm. In this example there was not.

Artificial Intelligence (Elektor, May 1988)

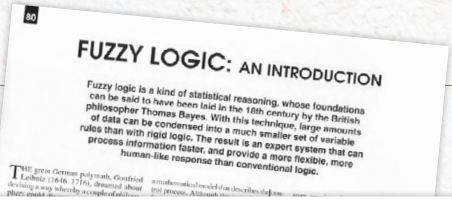
Focus: Discusses the evolution of artificial intelligence, from its Charles Babbage and Alan Turing to its development as a recognized field in 1956, while addressing the misconceptions and myths surrounding machine intelligence

Man is intelligent, but his intelligence is often thwarted (or worse) by his environment. That realization has given rise to a dream: that one day it may be possible to build a machine that can think, that is, need not be programmed to perform its functions. Machine intelligence was first thought of by Charles Babbage (1792-1871). Later, Alan Mathison Turing (1912-1954) achieved immortality through the Turing Machine, which purports to show that machines are mathematical objects, and his proof of the theorem that the set of mathematical tasks that is computable is exactly the same as the set that can be executed by the machine. He also formulated a theory about what questions could in principle be answered by an intelligent machine. Artificial intelligence grew out of the work on digital computers during the Second World War and was officially recognized as a branch of computing science in 1956. Since those early days, artificial intelligence has given rise to a number of myths, particularly, but not only, in the popular press. However, claims of computers achieving this and that, without human intervention, always prove, on close examination, to be mere illusions of intelligence. These illusions are created by the fact that computers work so extremely fast. Continue reading: *www.elektormagazine.com/magazine/elektor-198805/47340*

Intelligence, Intentionality and Self Awareness (Elektor, November 1989)

Focus: Discusses some of the problems in ascribing intelligence to computers, and investigates intentionality, the origin of self awareness, and brain and machine translations

In his article "Artificial Intelligence," M. Seymour provides an interesting and informative account of some of the problems met by computer designers in attempting to produce machines that exhibit artificial intelligence. The article discusses arguments for and against what constitutes artificial intelligence, including the existence or otherwise of intentionality (Searle, 1984). The present paper examines some of the concepts from the point of view of a psychologist, who was a student at Manchester when Alan Turing was working on the theoretical aspects of information processing. The power of electronic devices has increased enormously since that time, but perhaps there has not been a similar growth in defining the terminology used to describe computer activities and brain activities. At the simplest level there has been revival of anthropomorphism, a condemnatory appellation feared by biologists accused of reading human characteristics into the behaviour pattern of lower animals. However, equally imprecise use of language is exemplified by phrases such as 'computers talking to each other.' This is largely a matter of economy in the use of words, since it is easier to use concepts already in existence than to invent new ones, but there are dangers in over-extending the concepts to include things that are not justifiable. Continue reading: *www.elektormagazine.com/magazine/elektor-198911/47684*


80

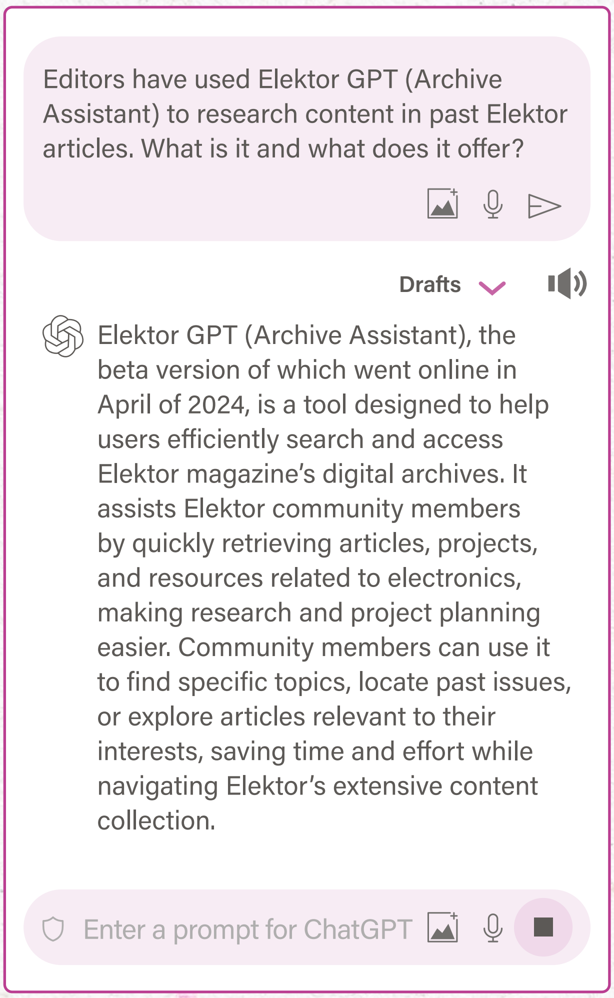
FUZZY LOGIC: AN INTRODUCTION

Fuzzy logic is a kind of statistical reasoning, whose foundations can be said to have been laid in the 18th century by the British philosopher Thomas Bayes. With this technique, large amounts of data can be condensed into a much smaller set of variable rules than with rigid logic. The result is an expert system that can process information faster, and provide a more flexible, more human-like response than conventional logic.

Fuzzy Logic: An Introduction (Elektor, July 1992)

Focus: Introduces fuzzy logic as a flexible reasoning method suitable for expert systems and human-like responses in AI

Fuzzy logic is a kind of statistical reasoning, whose foundations can be said to have been laid in the 18th century by the British philosopher Thomas Bayes. With this technique, large amounts of data can be condensed into a much smaller set of variable rules than with rigid logic. The result is an expert system that can process information faster, and provide a more flexible, more human-like response than conventional logic. The great German polymath, Gottfried Leibniz (1646-1716), dreamed about devising a way whereby a couple of philosophers could discuss and settle any human argument once and for all by pure logic. But he and many other thinkers after him have discovered that there are many problems that cannot be solved by just logic. This realization gave rise to another way of attempting to solve problems: the use of statistics. In statistical reasoning, probabilities express the idea of 'perhaps.' One method of statistical reasoning, whose foundations can be said to have been laid in the 18th century by the British philosopher Thomas Bayes, is called fuzzy logic. In fuzzy logic, there is not just 'true' and 'false,' 1s and 0s, 'black' and 'white,' but also all the various grades of grey in between. Fuzzy logic can condense large amounts of data into a much smaller set of variable rules than rigid logic. The result is an expert system that can process information faster, and provide a more flexible, more human-like response than conventional logic. Continue reading: *www.elektormagazine.com/magazine/elektor-199207/32683*


Editors have used Elektor GPT (Archive Assistant) to research content in past Elektor articles. What is it and what does it offer?

Elektor GPT (Archive Assistant), the beta version of which went online in April of 2024, is a tool designed to help users efficiently search and access Elektor magazine's digital archives. It assists Elektor community members by quickly retrieving articles, projects, and resources related to electronics, making research and project planning easier. Community members can use it to find specific topics, locate past issues, or explore articles relevant to their interests, saving time and effort while navigating Elektor's extensive content collection.

COMPUTER

Neural Networks in Control

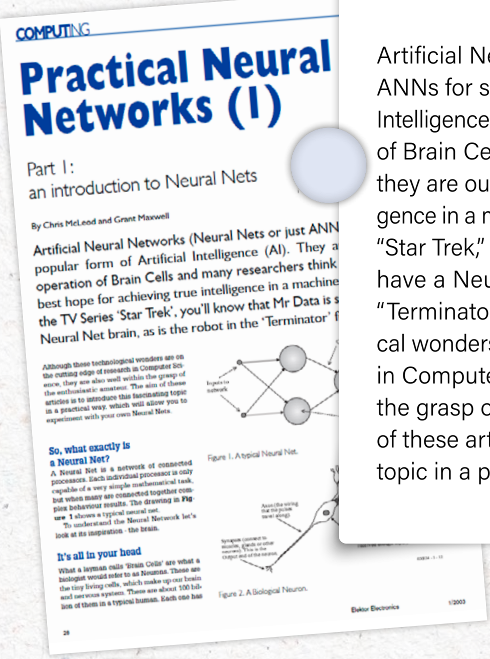
neurocomputers are here

By Owen Bishop

Neural Networks in Control: Neurocomputers Are Here (Elektor, October 2001)

Focus: Discusses neurons, the possibilities of implementing the neurocomputer, modeling an artificial neural network (ANN), and training

Neurons are usually organised into well-defined structures, such as the brain, the spinal cord and the peripheral nerves. Each neuron is a centre of information transfer (Note: transfer, not storage). It can receive information as electrical signals from a number of other neurons in the network, process this information, and pass it on to other neurons. In the human brain, for instance, a neuron connects to as many as 10,000 other neurons. The extent to which information is transferred from neuron to neuron depends on the number of physical connections existing between the neurons and the intensity (or strength) of each connection. The number of possible connections is far greater than the number of neurons present. We estimate that a human brain may contain as many as 1016 connections, an enormous resource for memory and intelligent action... Given that the neural network of an animal is capable of complex recognition and control actions, researchers have tried to build the artificial equivalent, an artificial neural network or ANN. An ANN consists of neuron units mimicking the action of animal neurons. They have input, processing and output stages. The earliest electronic ANNs were based on valve circuitry, but little progress was made until the advent of VLSI. Now, with the enormous computing power of even the smallest of bench-top computers, we model ANNs mathematically. This can be done in a system based on conventional microprocessors, as illustrated by the software described later. There are also custom-designed processors used to build true neurocomputers. These are designed to excel in the particular kinds of processing that are needed to model ANNs, which is done most efficiently by using parallel processing. Continue reading: *www.elektormagazine.com/magazine/elektor-200110/17112*


COMPUTING

Practical Neural Networks (I)

Part 1:
an introduction to Neural Nets

By Chris McLeod and Grant Maxwell

Artificial Neural Networks (Neural Nets or just ANN
popular form of Artificial Intelligence (AI). They a
operation of Brain Cells and many researchers think
best hope for achieving true intelligence in a machine
the TV Series 'Star Trek', you'll know that Mr Data is s
Neural Net brain, as is the robot in the 'Terminator' f

Although these technological wonders are on the cutting edge of research in Computer Science, they are also well within the grasp of the enthusiastic amateur. The aim of these articles is to introduce this fascinating topic in a practical way, which will allow you to experiment with your own Neural Nets.

So, what exactly is a Neural Net?

A Neural Net is a network of connected processors. Each individual processor is only capable of a very simple mathematical task, but when many are connected together complex behaviour results. The drawing in Figure 1 shows a typical neural net.

To understand the Neural Network let's look at its inspiration - the brain.

It's all in your head

What a layman calls 'Brain Cells' are what a biologist would refer to as Neurons. These are the tiny living cells, which make up our brain and nervous system. There are about 100 billion of them in a typical human. Each one has

Figure 1. A typical Neural Net.

Figure 2. A Biological Neuron.

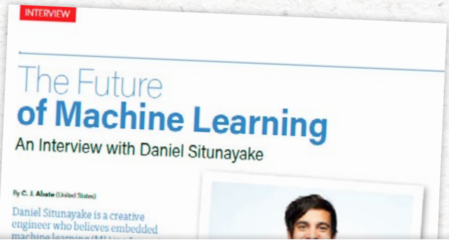
28

Elektor Electronics 1/2003

Practical Neural Networks (Elektor, January 2003)

Focus: Covers neural networks, the birth and history of the ANN, and learning and training algorithms

Artificial Neural Networks (Neural Nets or just ANNs for short) are a popular form of Artificial Intelligence (AI). They are based on the operation of Brain Cells and many researchers think that they are our best hope for achieving true intelligence in a machine. If you're a fan of the TV series "Star Trek," you'll know that Data is supposed to have a Neural Net brain, as is the robot in the "Terminator" films. Although these technological wonders are on the cutting edge of research in Computer Science, they are also well within the grasp of the enthusiastic amateur. The aim of these articles is to introduce this fascinating topic in a practical way, which will allow you to experiment with your own Neural Nets. A Neural Net is a network of connected processors. Each individual processor is only capable of a very simple mathematical task, but when many are connected together complex behaviour results. The drawing in Figure 1 shows a typical neural net. To understand the Neural Network let's look at its inspiration - the brain. What a layman calls 'Brain Cells' are what a biologist would refer to as Neurons. These are the tiny living cells, which make up our brain and nervous system. There are about 100 billion of them in a typical human. Continue reading: *www.elektormagazine.com/magazine/elektor-200301/17413*


INTERVIEW

The Future of Machine Learning

An Interview with Daniel Situnayake

By C. J. Abate (United States)

Daniel Situnayake is a creative engineer who believes embedded

The Future of Machine Learning (Elektor, November/December 2020)

Focus: Discusses embedded machine learning and introduces TinyML, Edge Impulse, and TinyML projects

Daniel Situnayake is a creative engineer who believes embedded machine learning (ML) is a "once-in-a-generation technology." Here he talks about ML's potential and introduces TinyML, as well as a few ideal applications. We also touch on his experience as a developer at Google. Situnayake states: "There are four main constraints that make an application ideal for TinyML. If an application has limited connectivity, low tolerance for latency, a limited source of power, or a requirement for strong privacy, TinyML can potentially help, by avoiding the need to send data from the device. It's worth noting that only certain problems can be solved by machine learning. They need to be problems where a bit of fuzziness is OK, since ML can rarely give absolute answers. It's perfect for taking animal photos, where the occasional mistake is acceptable, but it might not be reliable enough in some safety-critical applications." Continue reading: *www.elektormagazine.com/magazine/elektor-159/59100*

TRAINING

IMAGE PROCESSING with the Nvidia

Image Processing with the Nvidia Jetson Nano (Elektor, September/October 2021)

Focus: Covers the Nvidia Jetson Nano, training with the Jetson Nano, development via a web browser, and more

No developer should ignore the potential of artificial intelligence (AI). Tools like the Sipeed Maixduino and the Maxim MAX78000, which we have already presented in Elektor, are suitable options for taking your first steps into the world of AI. The Nvidia Jetson Nano is a much more powerful hardware from the class of single board computers, equipped with a quad-core ARM processor, 4 GB RAM and a GPU with 128 CUDA cores. In this series of articles, I show you which applications are possible, for example, in the fields of image processing or robotics… The Nvidia Jetson Nano is a much more powerful piece of hardware from the single board computer (SBC) class. Equipped with a quad-core ARM processor, 4 GB RAM and a GPU with 128 CUDA cores, the Jetson Nano is a good platform for investigating AI and neural networks. Applications range from speech recognition to classifying objects in captured video images to full-blown robot control. Continue reading: www.elektormagazine.com/magazine/elektor-183/59829

Related Product

- **Elektor Archive (USB Stick)**
  www.elektor.com/20751

WEB LINK

[1] Elektor GPT (ChatGPT registration necessary):
https://tinyurl.com/Elektor-GPT

240572-01

Hailo AI Kit
with Raspberry Pi 5

By Dogan Ibrahim (United Kingdom)

For those of you familiar with the Raspberry 5 but hesitant about Edge Computing and relevant hardware, Elektor's book, *Getting Started with the Raspberry Pi 5 AI Kit*, is a fantastic stepping stone. Packed with examples for a flying start to Raspberry 5-based AI programming, the book has the Elektor Raspberry Pi AI Kit as a solid base for the hardware. To show the potential of the book and kit jointly, here's an excerpt demonstrating person recognition in two ways with free AI tools, plus a brief introduction to Hailo Software Components.

Editor's Note. *This article is an excerpt from the Elektor book* Getting Started with the Raspberry Pi AI Kit, *which will be published by Elektor in 2024. It was formatted and lightly edited to match* ElektorMag's *conventions and page layout. The author and editor are happy to help with queries. Contact details are in the* **Questions or Comments?** *box.*

The Raspberry Pi 5 AI Kit (**Figure 1**) is a 13 TOPS (trillion of operations per second) neural network inference accelerator built around the Hailo-8L chip. The AI Kit includes the Raspberry Pi M.2 HAT+ and a Hailo AI acceleration module for use with the Raspberry Pi 5. The module uses the M.2 2242 form factor, and comes pre-installed in the M.2 HAT+, to which it connects through an M key edge connector. The M.2 HAT+ communicates between the AI module's M.2 interface and the Raspberry Pi 5's PCIe 2.0 interface. As a result, the AI kit provides an accessible, cost-effective, and power-efficient way, which can be used in many AI-based applications, such as object recognition, automation, process control, security, robotics, medical and many more.

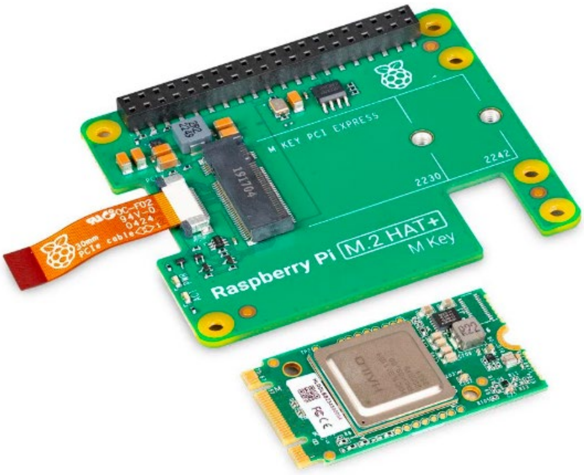


Figure 1: AI Kit disassembled.

Features

- Contains a neural network inference accelerator capable of 13 TOPS
- Fully integrated into Raspberry Pi's camera software stack
- Thermal pad pre-fitted between module and HAT+ spreads heat across components, improving performance
- Conforms to Raspberry Pi HAT+ specification

Included

Each AI Kit comes with a pre-installed AI module, ribbon cable, GPIO header, and mounting hardware. In detail, included with the kit are:

- Hailo 8L accelerator with Neural Processing Unit (NPU)
- Raspberry Pi M.2 HAT+
- Thermal pad pre-fitted between the module and the M.2 HAT+
- Mounting hardware kit (spacers, screws)

- 16 mm GPIO stacking header
- 4 threaded spacers
- 8 screws
- 1 knurled double-flanged drive attachment screw to secure and support the M.2 peripheral

Hardware Installation

Figure 2 shows the AI Kit with the Raspberry Pi 5. The AI kit must be installed on top of the Raspberry Pi 5 before it can be used. It is recommended to use the AI Kit with the Raspberry Pi active Cooler. Thus, if you have an Active Cooler, make sure that you install it before installing the AI Kit. Also, it is recommended to install the Raspberry Pi 5 camera (e.g., camera V3) before you install the AI Kit, as this way it will be easier to connect the camera cable.

Installing the AI Kit Software

First of all, you should ensure that your Raspberry Pi 5 is installed with the latest Bookworm operating system.

Run the following commands to update the operating system to latest version:

```
pi@raspberrypi:~ $ sudo apt update
pi@raspberrypi:~ $ sudo apt full-upgrade
```

Ensure that the Raspberry Pi 5 firmware is up to date:

```
pi@raspberrypi:~ $ sudo rpi-eeprom-update
```

If you see December 2023 or a later date, proceed to the next step. If you see a date earlier than 6 December 2023, run the following command to open the Raspberry Pi configuration tool:

```
pi@raspberrypi:~ $ sudo raspi-config
```

Under *Advanced Options Bootloader Version*, choose *Latest*. Then exit `raspi-config` with *Finish* or the *Escape* key. Run the following command:

```
pi@raspberrypi:~ $ sudo rpi-eeprom-update -a
```

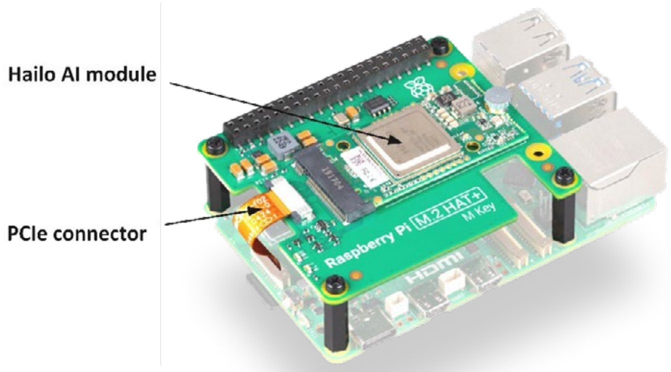


Figure 2: AI Kit assembled.

To achieve optimal performance from the Hailo device, it is necessary to set *PCIe* to *Gen3*. While using *Gen2* is an option, it will result in lower performance. The steps are:

```
pi@raspberrypi:~ $ sudo raspi-config
```

Select option 6. *Advanced Options*, then select option *A8 PCIe Speed*. Choose *Yes* to enable *PCIe Gen 3* mode. Click *Finish* to exit.

Reboot your Raspberry Pi 5:

```
pi@raspberrypi:~ $  sudo reboot
```

Installing the Hailo AI Kit Software

Run the following command to install the Hailo AI Kit software on Raspberry Pi 5:

```
pi@raspberrypi:~ $ sudo apt install hailo-all
```

The above command installs the following dependencies:

- Hailo kernel device driver and firmware
- HailoRT runtime software
- Hailo Tappas core package
- The rpicam-apps Hailo post-processing software demo stages

Reboot your Raspberry Pi 5 for the above settings to take effect:

```
pi@raspberrypi:~ $  sudo reboot
```

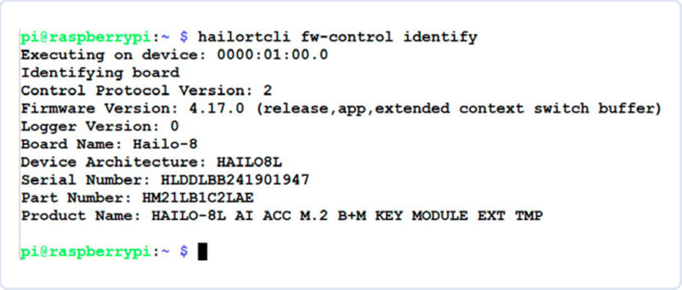
Verifying the Software Installation

You should verify the software installation to make sure that all the software components are in place and the AI kit can be recognized by the Raspberry Pi 5. The steps are:

- Make sure the AI kit hardware is installed on your Raspberry Pi 5 as described earlier
- Enter the following command:

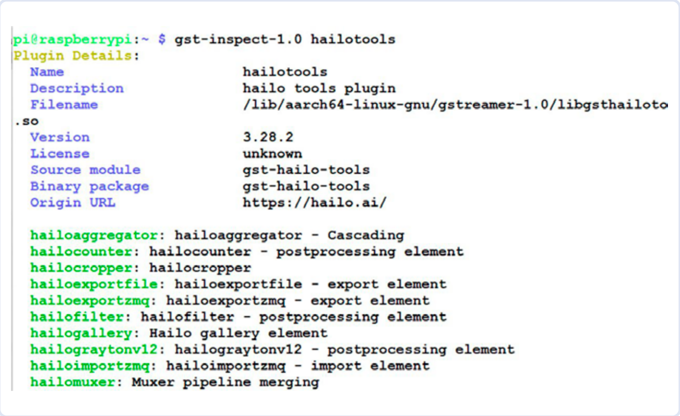
```
pi@raspberrypi:~ $ hailortcli fw-control identify
```

If everything is ok, you should see an output similar to **Figure 3**. Note here that the software version is 4.17.



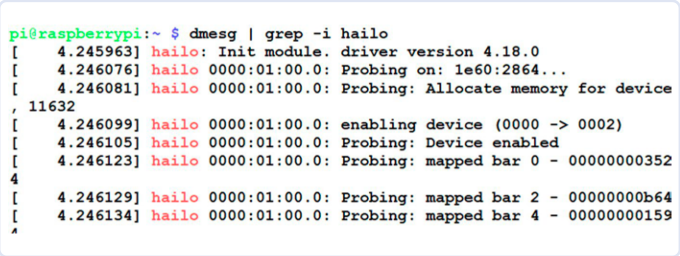
```
pi@raspberrypi:~ $ hailortcli fw-control identify
Executing on device: 0000:01:00.0
Identifying board
Control Protocol Version: 2
Firmware Version: 4.17.0 (release,app,extended context switch buffer)
Logger Version: 0
Board Name: Hailo-8
Device Architecture: HAILO8L
Serial Number: HLDDLBB241901947
Part Number: HM21LB1C2LAE
Product Name: HAILO-8L AI ACC M.2 B+M KEY MODULE EXT TMP

pi@raspberrypi:~ $ █
```

Figure 3: Verifying the software installation.



```
pi@raspberrypi:~ $ gst-inspect-1.0 hailotools
Plugin Details:
  Name                     hailotools
  Description              hailo tools plugin
  Filename                 /lib/aarch64-linux-gnu/gstreamer-1.0/libgsthailoto
.so
  Version                  3.28.2
  License                  unknown
  Source module            gst-hailo-tools
  Binary package           gst-hailo-tools
  Origin URL               https://hailo.ai/

  hailoaggregator: hailoaggregator - Cascading
  hailocounter: hailocounter - postprocessing element
  hailocropper: hailocropper
  hailoexportfile: hailoexportfile - export element
  hailoexportzmq: hailoexportzmq - export element
  hailofilter: hailofilter - postprocessing element
  hailogallery: Hailo gallery element
  hailograytonv12: hailograytonv12 - postprocessing element
  hailoimportzmq: hailoimportzmq - import element
  hailomuxer: Muxer pipeline merging
```

Figure 4: Test TAPPAS Core installation (part of the display is shown).



```
pi@raspberrypi:~ $ dmesg | grep -i hailo
[    4.245963] hailo: Init module. driver version 4.18.0
[    4.246076] hailo 0000:01:00.0: Probing on: 1e60:2864...
[    4.246081] hailo 0000:01:00.0: Probing: Allocate memory for device
, 11632
[    4.246099] hailo 0000:01:00.0: enabling device (0000 -> 0002)
[    4.246105] hailo 0000:01:00.0: Probing: Device enabled
[    4.246123] hailo 0000:01:00.0: Probing: mapped bar 0 - 00000000352
4
[    4.246129] hailo 0000:01:00.0: Probing: mapped bar 2 - 00000000b64
[    4.246134] hailo 0000:01:00.0: Probing: mapped bar 4 - 00000000159
```

Figure 5: Checking kernel logs (part of the display is shown).

- You should also test for the *TAPPAS Core installation* by running the following command:

```
pi@raspberrypi:~ $ gst-inspect-1.0 hailotools
```

If everything is ok, you should see something similar to **Figure 4**.

If `hailo` or `hailotools` are not found, try deleting the GStreamer registry:

```
pi@raspberrypi:~ $  rm ~/.cache/gstreamer-1.0/registry.
aarch64.bin
```

- Additionally, you can run the following command to check the kernel logs:

```
pi@raspberrypi:~ $  dmesg | grep -i hailo
```

Which should give an output similar to **Figure 5**.

- Finally, you should check to make sure that the camera operates correctly. Enter the following command, which will show a camera preview for 10 seconds.

```
pi@raspberrypi:~ $  rpicam-hello -t 10s
```

If you are using a V3 camera module with your Raspberry Pi 5, add the following lines to your *config.txt* file:

- Edit *config.txt*: `sudo nano /boot/firmware/config.txt`
- Change line: `camera_auto_detect=1` to `camera_auto_detect=0`
- Insert line: `dtoverlay=imx708`
- Type *CntrlX* followed by *Y* to save and exit

Re-enter the command `rpicam-hello -t 10s` to check the camera operation again. At this point, you have verified that the AI Kit hardware and software have been installed correctly, and your camera is operating.

Running the Demo Programs

A number of demo programs are available with the AI Kit for the Raspberry Pi 5. The steps to install some of these demo programs are:

- Clone the repository

```
git clone https://github.com/hailo-ai/hailo-rpi5-
examples.git
```

- Enter the repository directory:

```
cd hailo-rpi5-examples
```

- To run the examples, you should ensure your environment is set up correctly. You can set it all up by sourcing the following script. This script will set the required environment variables and activate Hailo virtual environment (if it doesn't exist, it will create it).

```
source setup_env.sh
```

- Make sure you are in the virtual environment and run the following command. This may take some time, wait until it is finished.

```
pip install -r requirements.txt
```

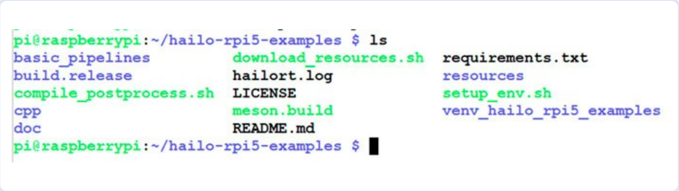
- Resource download. This may take some time, wait until it is finished:

```
./download_resources.sh
```

This repository contains examples of basic pipelines for Hailo's RPi5 platform. The examples demonstrate object detection, human pose estimation, and instance segmentation. It is built as templates to allow you to use these applications as a basis for your own projects.

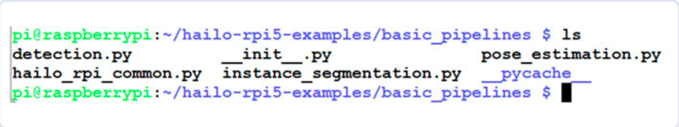
The *hailo-rpi5-examples* folder contains the folders and files shown in **Figure 6**.

The applications programs (detection, pose estimation, instance segmentation) are in folder *basic_pipelines* as shown in **Figure 7**.



```
pi@raspberrypi:~/hailo-rpi5-examples $ ls
basic_pipelines         download_resources.sh  requirements.txt
build.release           hailort.log            resources
compile_postprocess.sh  LICENSE                setup_env.sh
cpp                     meson.build            venv_hailo_rpi5_examples
doc                     README.md
pi@raspberrypi:~/hailo-rpi5-examples $ █
```

Figure 6: Examples folder contents.



```
pi@raspberrypi:~/hailo-rpi5-examples/basic_pipelines $ ls
detection.py         __init__.py                pose_estimation.py
hailo_rpi_common.py  instance_segmentation.py   __pycache__
pi@raspberrypi:~/hailo-rpi5-examples/basic_pipelines $ █
```

Figure 7: Applications programs.

Detection Demo Example

This example demonstrates object detection. It uses *YOLOv6n* model as default. It supports also *yolov8s* and *yolox_s_leaky* models. It uses Hailo's NMS (Non-Maximum Suppression) layer as part of the HEF file, so all detection networks which are compiled with NMS can be used with the same code.

To run the example, use (make sure you are in Desktop GUI mode):

```
cd hailo-rpi5-examples
source setup_env.sh
python basic_pipelines/detection.py --input resources/
detection0.mp4
```

This application reads the *detection0.mp4* movie file and detects and indicates various items in the file. **Figure 8** shows the output of the program displayed on the Raspberry Pi monitor, where the identified objects are shown with rectangles.

To close the application press *Ctrl+C*. For additional input options, enter the following command:

```
python basic_pipelines/detection.py --help
```

For example, to read images from a Raspberry Pi 5 camera, enter `--input rpi`. The following command will read the images from the Raspberry Pi 5 camera:

```
python basic_pipelines/detection.py --input rpi
```

Also, to read images from a USB camera (webcam), enter `--input /dev/video0`.

Note that the USB camera is not guaranteed to be `/dev/video0`. You can check which video devices are available by entering the command:

```
ls /dev/video*
```

or, enter the command:

```
v4l2-ctl -list-devices
```

You can test whether the USB camera is working by entering (Note that it is lowercase L, not 1):

```
ffplay -f v4l2 /dev/video0
```

If you get an error try another device (e.g. `/dev/video2`). In author's setup the USB camera was named as `/dev/video8`. Your USB camera will probably have a different device name.

For example, to read images from the USB camera, the author used the option `-i /dev/video8`. In the following example, a keyboard is detected using a USB camera:

```
python basic_pipelines/detection.py -i /dev/video8
```

The output is shown in **Figure 9** with the keyboard label displayed at the top left corner of the display.

To print the frame rate add the `--print-fps` flag. This will print the frame rate to the terminal and to the video output window.

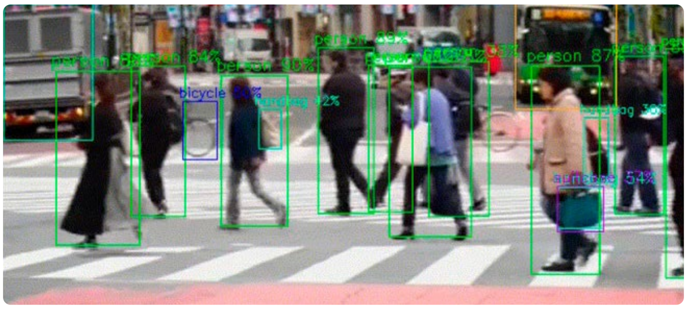


Figure 8: Output of the program.

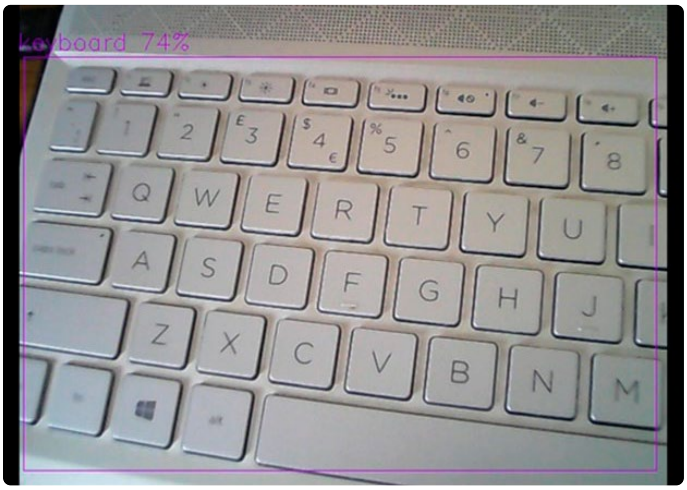


Figure 9: Detecting a keyboard.

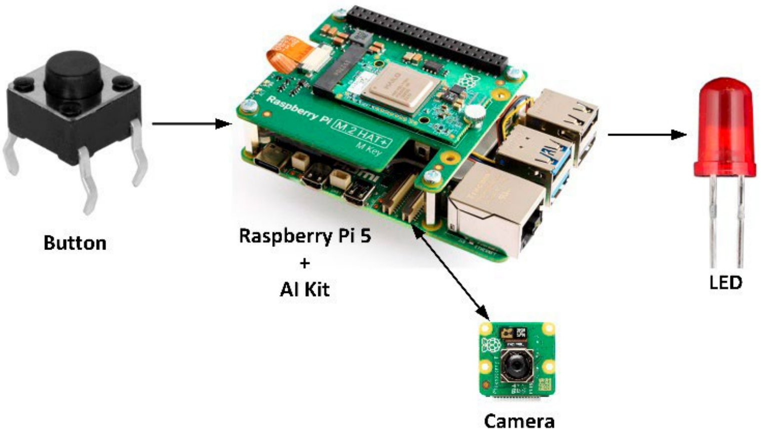


Figure 10: Block diagram.

Project — Detect Person Presence Using the AI Kit — LED Output When Person Detected

Description: In this project, a person's presence is detected, and an LED is turned ON when a person is detected. The program waits with the LED ON until the user presses a reset button to acknowledge the detection. At this time, the LED turns OFF and the detection process starts again. This is a simple but useful example. The LED can be replaced with a relay or a buzzer if required. Also, the program can be modified to detect objects other than persons.

Block diagram: **Figure 10** shows the project block diagram.

Circuit diagram: The circuit diagram is shown in **Figure 11**. The LED is connected to port pin 21 of Raspberry Pi 5 through a 330-Ω current limiting resistor. The pushbutton is connected to port pin 20. The default state of the button is at logic 1 and goes to logic 0 when the button is pressed.

Program listing: The Python program listing is shown in **Listing 1**. The program creates a subprogram and runs as a shell program. At the beginning of the program, the *gpiozero* library is imported and Raspberry Pi 5 GPIO port 21 is assigned to `led` and `button` is assigned to port 20.

The *rpicam-apps* post-processing Hailo inference program is started in a while loop and the output of this program is compared if it contains the word *person*. i.e. If a person has been detected. If so, the message *Person detected* is displayed on the Raspberry Pi 5 monitor, and the shell program exits and the LED is turned ON. The program then waits until the button is pressed. Pressing the button is like an acknowledgement from the user. Pressing the button turns OFF the LED. The program then waits for 15 seconds, and then the detection process resumes from the beginning.

This program can easily be modified by replacing the LED with a relay or a buzzer, and changing the waiting time to suit the required application. One possible application of this project is in security, where it may be required to detect when a person is sensed in-front of the camera. Also, other objects can be detected and then required actions can be taken.

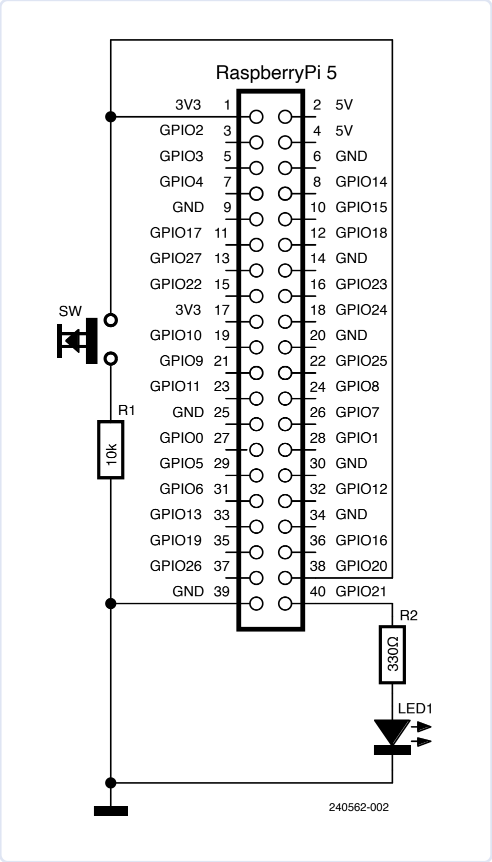


Figure 11: Circuit diagram.

Object Detection Training

AI-based object detection and training is a highly complex process and requires good understanding and working knowledge of programming concepts, especially Python programming. Chapter 11 of the book describes in detail how to carry out object detection training using an Ubuntu-based computer, where the output is loaded to a Raspberry Pi 5.

240562-01

About the Author

Prof. Dr. Dogan Ibrahim has a BSc degree in electronic engineering, an MSc degree in automatic control engineering, and a PhD degree in digital signal processing. Dogan has worked in many industrial organizations before he returned to academic life. Prof. Ibrahim is the author of over 70 technical books and has published over 200 technical articles on microcontrollers, microprocessors, and related fields. He is a Chartered electrical engineer and a Fellow of the Institution of the Engineering Technology. He is a certified Arduino professional.

Listing 1: Python program.

```
#-----------------------------------------------------------------
#          DETECT PERSON AND TURN ON LED
#          ---------------------------------------------------
#
# File  :  Detect.py
# Date  : October, 2024
# Author: Dogan Ibrahim
#----------------------------------------------------------------
import os
import subprocess
from gpiozero import LED, Button
from time import sleep
led = LED(21)                                  # LED port
button = Button(20)                            # Button port
led.off()                                      # LED OFF
while(True):
   p=subprocess.call("rpicam-hello -n -v 2 -t 0 \
     --post-process-file /home/pi/rpicam-apps/assets/hailo_yolov8_inference.json\
     --lores-width 640 --lores-height 640 2>&1 | grep -m 1 -c \
     --line-buffered 'person'",shell=True)
     print("Person detected...")
#
# At this point a person has been detected
#
   led.on()
   button.wait_for_press()                   # Wait for button press
   led.off()
   sleep(15)
```

Questions or Comments?

Do you have any questions or comments related to this article? Email the author at d.ibrahim@btinternet.com or Elektor at editor@elektor.com.

Related Products

- **Dogan Ibrahim, *Getting Started with the Raspberry Pi AI Kit* (Elektor 2024)**
  www.elektor.com/21031
- **Dogan Ibrahim, *Getting Started with the Raspberry Pi AI Kit* (Elektor 2024, E-book)**
  www.elektor.com/21032
- **Raspberry Pi AI Kit**
  www.elektor.com/20879

AI's Impact on Smart Building Energy Management

As artificial intelligence technology continues to advance, its impact on energy consumption and carbon emissions in buildings could be profound [1]. This research [2] explores how AI can reduce energy costs and emissions, emphasizing the need for further studies to quantify AI's full potential. The study proposes a systematic approach to evaluate AI's impact, predicting significant reductions in energy use and emissions by 2050.

With AI technology and efficiency policies, CO_2 emissions could be cut by around 40% by 2050 compared to maintaining the current trajectory without any intervention (BAU). Implementing both AI and policy-driven efforts could lead to a 60% reduction. By further incorporating low-emission power generation, emissions could be nearly eliminated by 2050, achieving savings of up to 93–95% compared to current trends.

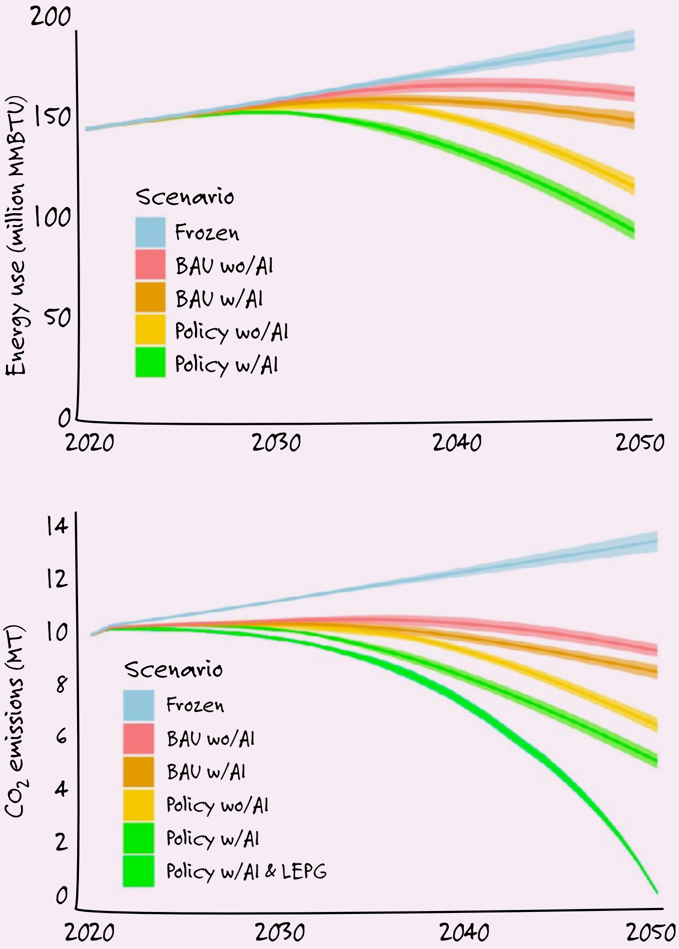
AI could help decrease CO_2 emissions by:

~40%

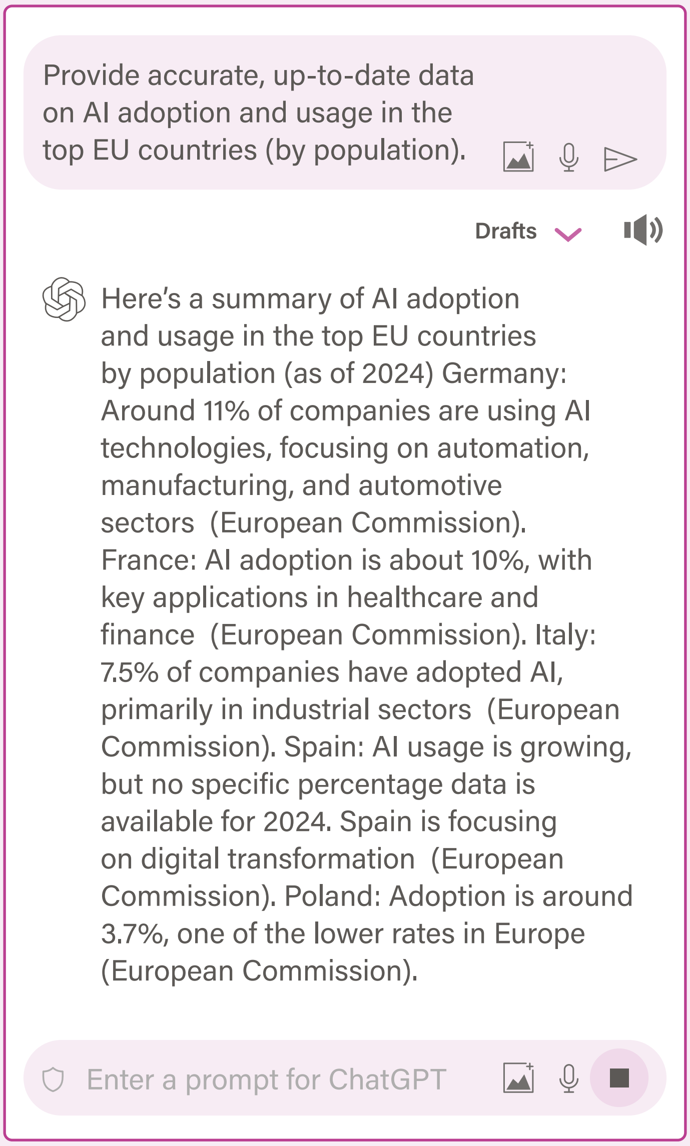
compared to the BAU scenario.

~90%

from the BAU scenario when combined with energy efficiency policies and low-emission power generation (LEPG).



Energy consumption by scenario. BAU stands for business-as-usual. The darker color for each scenario indicates the average estimate and the lighter color for each scenario indicates the estimated ranges from the sensitivity analysis.[2]



Provide accurate, up-to-date data on AI adoption and usage in the top EU countries (by population).

Here's a summary of AI adoption and usage in the top EU countries by population (as of 2024) Germany: Around 11% of companies are using AI technologies, focusing on automation, manufacturing, and automotive sectors (European Commission). France: AI adoption is about 10%, with key applications in healthcare and finance (European Commission). Italy: 7.5% of companies have adopted AI, primarily in industrial sectors (European Commission). Spain: AI usage is growing, but no specific percentage data is available for 2024. Spain is focusing on digital transformation (European Commission). Poland: Adoption is around 3.7%, one of the lower rates in Europe (European Commission).

Tech Giants Developing AI Chips

Due to the global chip shortage caused by the COVID-19 pandemic in 2020, the production of GPUs was severely impacted. The pandemic disrupted the global supply chain, resulting in delays in chip production and delivery. As a result, several companies have started creating their own AI chips. Here's a list of those companies:

- Google: Improving its Tensor Processing Unit (TPU) [4] technology for Google Cloud.
- OpenAI: Exploring the development of proprietary AI chips [5].
- Microsoft: Launched two custom chips — Microsoft Azure Maia AI Accelerator and Azure Cobalt CPU [6].
- Amazon: Released the Inferentia AI chip [7] and second-gen AWS Trainium ML accelerator [8].
- Meta: Developing a custom chip for its AI initiatives [9].
- Huawei: Prioritizing AI chip production, slowing down smartphone production [10].
- Tesla: Created the Dojo AI chip [11] for machine learning in autonomous driving.
- Baidu: Developed the Kunlun AI chip for deep learning and speech recognition.
- Alibaba: Released the Hanguang 800 AI chip [12] to enhance ML in cloud services.

240435-01

AI Coding Assistant Adoption and Ongoing Challenges

In a recent survey conducted by Stack Overflow [3], it was found that developers are widely adopting AI coding assistants.

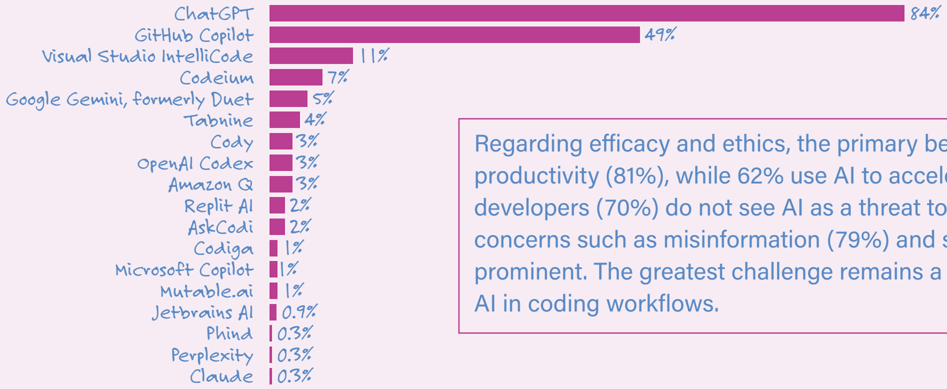
76% vs **70%**

of respondents either use or plan to use AI coding assistants from last year.

62% vs **44%**

now actively use AI tools in the previous year.

What is the primary code assistant tool professional developers use?



Regarding efficacy and ethics, the primary benefit mentioned is increased productivity (81%), while 62% use AI to accelerate their learning. Most developers (70%) do not see AI as a threat to their jobs, although ethical concerns such as misinformation (79%) and source attribution (65%) are prominent. The greatest challenge remains a lack of trust and understanding of AI in coding workflows.

WEB LINKS

[1] N. Flaherty, "AI can cut smart building energy management by over 19%," eeNews, 2024: https://tinyurl.com/ai-smart-energy
[2] C. Ding, J. Ke, M. Levine et al., "Potential of artificial intelligence in reducing energy and carbon emissions of commercial buildings at scale," Nat Commun (2024): https://doi.org/10.1038/s41467-024-50088-4
[3] "2024 Developer Survey," Stack Overflow, May 2024: https://survey.stackoverflow.co/2024/ai/
[4] Accelerate AI development with Google Cloud TPUs: https://cloud.google.com/tpu?hl=en
[5] A. Tong, M. Cherney, C. Bing, S. Nellis, "Exclusive: ChatGPT-owner OpenAI is exploring making its own AI chips," Reuters, 2023: https://tinyurl.com/chatgpt-chip
[6] J. Siegel, "With a systems approach to chips, Microsoft aims to tailor everything 'from silicon to service' to meet AI demand," Microsoft, 2023: https://tinyurl.com/microsoft-ai-chip
[7] AWS Inferentia: https://aws.amazon.com/machine-learning/inferentia/
[8] AWS Trainium: https://aws.amazon.com/machine-learning/trainium/
[9] F. Potkin, J. Zhu, "aExclusive: AI chip demand forces Huawei to slow smartphone production," Reuters, 2024: https://tinyurl.com/huawei-chip
[10] K. Paul, S. Nellis, M. Cherney, "Exclusive: Meta to deploy in-house custom chips this year to power AI drive," Reuters, 2024: https://tinyurl.com/meta-chip
[11] "TSMC Reportedly Commences Production of Tesla's Next-Generation Dojo Chips, Anticipates 40x Increase in Computing Power in 3 Years," TrendForce, 2024: https://tinyurl.com/tesla-chip
[12] Y. Jiao, L. Han, X. Long, et al.,"Hanguang 800 NPU," Alibaba Group: https://tinyurl.com/alibaba-chip

Elettronica In
www.elettronicain.it

AI-Based Universal IR Remote Control

Gesture-Controlled with MediaPipe and Raspberry Pi

By Antonio Aloisio (Italy)

If you are looking for a project to start experimenting with AI, this might be the right opportunity! With the help of a Raspberry Pi 5, Google's MediaPipe Studio platform and a tiny plug-in interface board, you may get a versatile gesture recognition system to control your TV or other devices operated through IR remote controls.

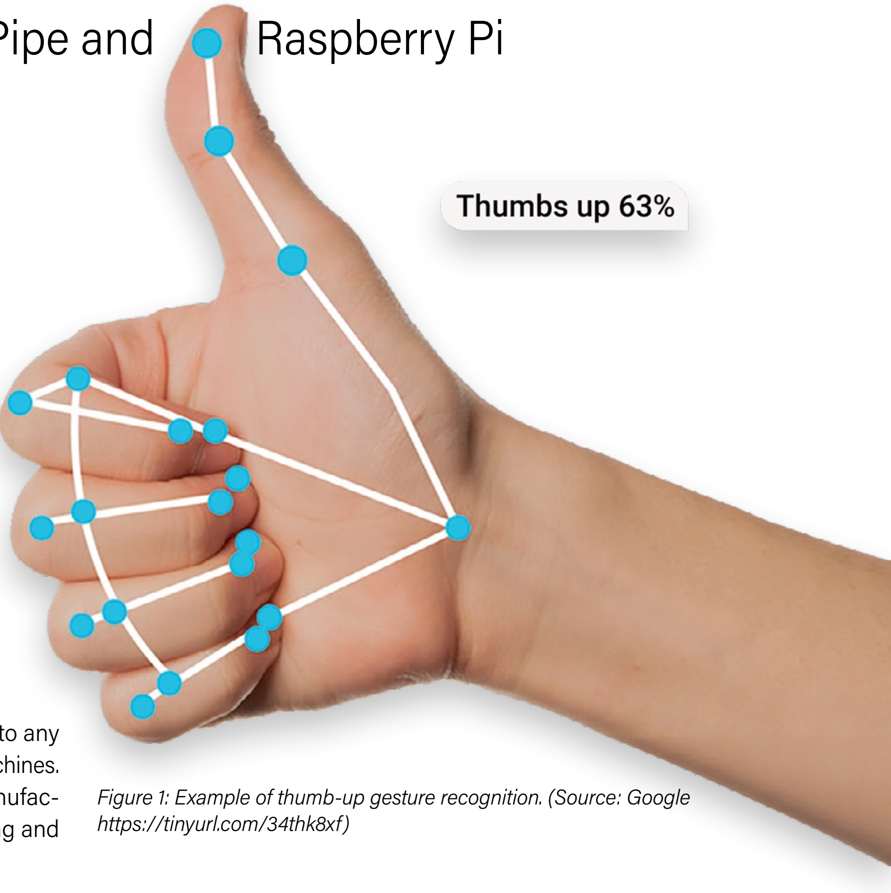


Figure 1: Example of thumb-up gesture recognition. (Source: Google https://tinyurl.com/34thk8xf)

HMI, short for *Human-Machine Interface*, is a term that refers to any technology or device capable of making humans interact with machines. HMIs are critical in a wide range of areas, from industrial manufacturing to consumer electronics, and are essential for simplifying and optimizing interactions between users and machines.

The most commonly used HMIs are:

- **Graphical User Interfaces (GUIs)**: are the most common and include graphical screens that allow users to interact with machines through icons, buttons and menus, typical of PC operating systems, for example.
- **Voice user interfaces (VUIs)**: allow users to interact with machines through voice commands. This type of HMI is becoming increasingly popular with the advent of voice assistance such as Siri by Apple, Google Assistant, and Amazon Alexa.
- **Touch user interfaces**: allow users to interact with machines through physical touch, such as touchscreens, keyboards and mice.
- **Gestural user interfaces**: allow users to interact with machines through physical movements, such as hand gestures detected by cameras or motion sensors, which will be the focus of this project.

Applications range from industrial automation (used to monitor and control machines and industrial processes) to consumer electronics (touchscreens are common in consumer devices such as smartphones, tablets, computers, and TV sets). In home automation systems, HMIs allow users to control lights, thermostats, security devices, and other smart home devices. They are critical in vehicles to enable users to interact with navigation systems, entertainment systems and other features.

Interaction must follow dictates of ergonomics and functionality that are not always achieved in all HMIs; just think of the recent criticism by Euro NCAP against the complexity of vehicles that have shifted key controls to center screens, forcing drivers to take their eyes off the road, increasing the risk of distraction accidents (source: ETSC [1]). Making HMIs more intuitive (user-friendly) is therefore one of the goals that companies are increasingly trying to achieve.

Interaction with machines is then an area we deal with daily, and has become crucial in a wide range of industries, helping to improve efficiency, safety and the overall user experience. Today, many of the HMI systems take advantage of artificial intelligence (AI) for their operation.

The project we propose in this article uses hand gesture recognition to give commands to a television set. We will use an AI system to replace one of the most common and "comforting" devices in everyday life, the remote control. At the end of the project, you'll have a lot of fun turning on the TV set with the "Jedi force" of palm opening!

The approach of the project is playful, but it is intended to be a gateway to the use of AI systems by everyone, in this case aimed at creating gestural HMI.

Software

AI is a field of computer science that focuses on developing systems and technologies that can simulate human intelligence processes, i.e., think and solve problems that we humans are used to. Machine learning (ML), on the other hand, is a fundamental subset of AI that studies how to make a computer system learn from past data to make predictions or decisions, without being explicitly programmed to do so.

The first step in ML is the acquisition of data, which can be of various types, such as text, images, sounds, or numbers. They can come from sensors, databases, or even social media. The importance of quality data is critical in determining the success of an ML model. ML algorithms, on the other hand, are the tools that enable computer systems to learn from data. There are various types of algorithms, each of which is suited to specific tasks. For example, regression algorithms can be used to make predictions based on historical data, while classification algorithms are useful for assigning categories to data.

Next comes the process of training an ML system (algorithm + data), which involves feeding the algorithm with labeled data; that is, data that already has correct answers associated with it. The algorithm then uses this data to learn the relationships. This step requires a certain amount of time and computational resources, but it is critical to the creation of accurate models.

Once trained, the model must be evaluated to determine how well it performs on data that are not present in it. This step is crucial to understand whether the model can generalize well beyond the data with which it was trained. If the model does not meet the expected performance, it may need to be optimized by adjusting various parameters or using tuning techniques. This process is known as model optimization and may require some experimentation to find the optimal combination of parameters.

MediaPipe

This big job of algorithm creation, training and optimization is provided as Open Source by Google with the mighty MediaPipe platform. The meaning of the term *MediaPipe* is related to the concept of *media pipeline*.

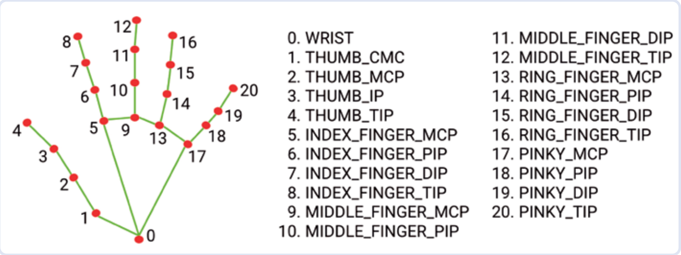


Figure 2: Recognized points on palm — landmarks. (Source: Google https://tinyurl.com/34thk8xf)

MediaPipe is defined by Google as *on-device*, meaning it is also capable of running on an offline hardware device, without the need for an internet connection. This makes our projects portable and integrable in various contexts. Google adds the term *no-code* to its platform, meaning it provides a system that requires little programming knowledge to achieve its goal.

A concept we are familiar with from the Arduino universe, which disclosed the world of electronics and code writing even to those who did not have the basics. MediaPipe also aims in this direction, proposing in a simple way an originally very complex topic like AI.

Let's see its potential at MediaPipe Studio gesture recognition webpage [2]. We need to log in with a Google account, allowing the browser to use the webcam. We frame our hand, it doesn't matter whether left or right, as it also identifies which one it is, and perform one of the following gestures:

1 - Fist
2 - Open palm
3 - Index finger pointing up
4 - Thumb pointing up
5 - Thumb down
6 - *Victory* symbol with two fingers
7 - *I love you* in the language of gestures (final position).

The page will show, in real time and overlaid on the video, 21 key palm points connected by colored segments, as visible in **Figure 1**. The key points are also referred to as *landmarks* and represent the significant parts of the hands, such as fingertips, thumbs, palms, and so on. Precise tracking of these landmarks provides insight into the pose and movements of hands within an application (**Figure 2**).

On the web page, below the video frame, there will also be a range of additional information, such as the percentage of confidence in having recognized the gesture performed and, of course, the type of gesture. MediaPipe then offers a solution called *Hand Tracking* that enables real-time hand recognition and tracking within a video stream. This module of MediaPipe has been named *HandLandmarks* accordingly. The module begins by detecting the presence of hands in the video frame and identifies the regions of the image that contain them. Once detected, it begins tracking hand landmarks.

With the landmarks identified, the module can estimate the pose of the hands within the video frame. This includes information such as hand orientation, finger direction, and other details that can be used to interpret gestures. MediaPipe is based on TensorFlow lite, i.e., the large ML platform (also from Google), but differs from it in that it is already pre-trained. For example, gesture recognition takes advantage

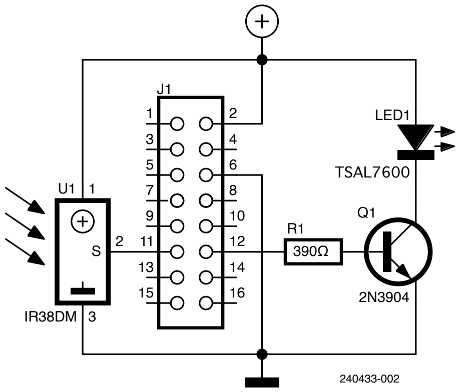


Figure 3: Schematic of the simple, plug-in IR Rx-Tx interface.

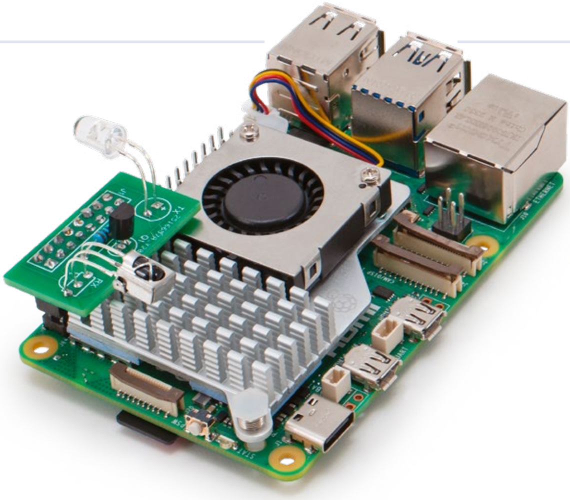


Figure 4: The small IR interface board, mounted on Raspberry Pi 5.

of about 30,000 images taken from the real world. You can also train the system yourself, with images of other gestures, by following the tutorial at [3].

Have fun exploring the other ML systems provided by MediaPipe such as image classification, object or face detection. The documentation therein will allow you to delve into the topics of most interest. MediaPipe is also multi-platform; in fact, you can create projects for Android, iOS and Web (JavaScript). In our project we will use Python, i.e. one of the most popular programming languages in the ML field (and beyond) for several reasons: its ease of use, the intuitive syntax — and the wide range of educational resources and libraries available — make it an excellent choice for beginners and programming experts alike.

One article would not even suffice as a primer for learning the big world of TensorFlow, MediaPipe and Python. But we would like you to use it to understand its capabilities and to enable you to direct your study of indispensable topics in present and future computing.

MediaPipe on Raspberry Pi

Let's start practically, by installing MediaPipe on our Raspberry Pi 5 so that we can use it on-device right away. Connect to your board as best you are used to, although we recommend VNC, for simplicity and power of use [4]. Python is native in the Raspberry Pi operating system. Take administrator rights with the `sudo bash` command and make sure you have everything updated with `apt update` and `apt upgrade`.

Install the MediaPipe library for Python with the terminal command `pip install mediapipe`. In Python, `pip` is a package management tool that makes it easy to install, manage, and update external libraries and frameworks.

The name *pip* is a recursive acronym that stands for *Python Installs Packages*. Some versions of the Raspberry Pi operating system may give a package installation error; if so, clear the external package management with the following command `rm /usr/lib/python3.*/EXTERNALLY-MANAGED`, and re-run `pip`.

You must now download the data model for MediaPipe gesture recognition at the link [5]. Place it in the same folder where you will create the Python program.

The second Python library we will use is *OpenCV* (Open Source Computer Vision Library), used for computer vision and image processing. OpenCV provides a wide range of features for image analysis and manipulation, pattern recognition (recurring patterns or regularities within a data set), object tracking, computer vision, and much more.

Let's install with `pip install opencv-python`. We have everything to try out on our stand-alone system (application or system that can run independently). If you are already a Python developer, use the editor you are used to — we used Geany — already installed on Raspberry OS. In **Listing 1,** you will find the program you need to write on the editor commented and explained in detail. Run it (airplane icon in Geany). A new window will then open with the live video stream and hand landmark. The recognized gesture will appear in the editor terminal.

Note how in the `match...case` section we have left as a comment (in Python, with the `#` symbol at the beginning of the line) the `os.system` instructions, i.e., those operating system commands that will allow sending the IR signals, which we will make operational after dealing with the hardware aspect.

Acquiring and Sending IR Signals

For learning and sending the infrared (IR) signals, we will make a simple interface — whose schematic is shown in **Figure 3** — with a tiny PCB housing, an IR receiver and an IR transmitter LED. With a convenient plug, it will directly plug into the Raspberry Pi GPIO header, as illustrated in **Figure 4**.

For image capture, we'll use a USB webcam connected to Raspberry Pi. The design is perfectly compatible with whatever cam you have for your board. To give support stability to the PCB, we used the first 16 pins (numbering from 1 → 3.3 V to 16 → GPIO 23), see **Figure 5**.

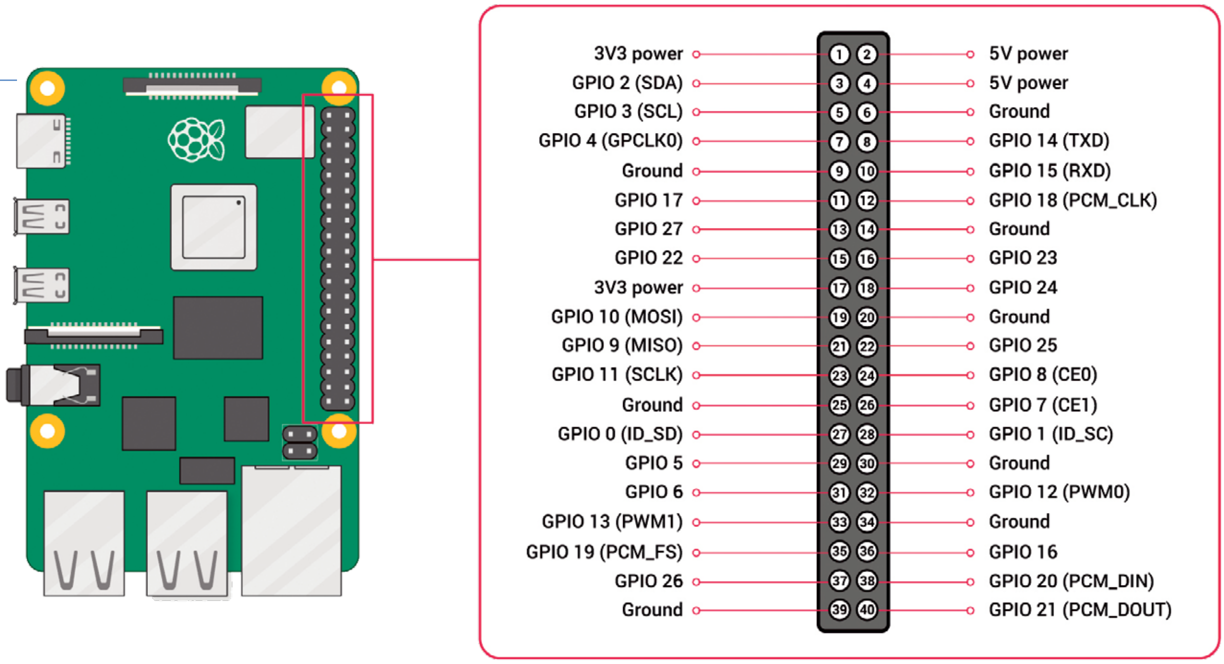


Figure 5: Raspberry Pi 5 GPIO's pinout. (Source: https://tinyurl.com/5dpvj5ne)

Actually connected are only four pins:

- Pin 2 → Vcc +5 V
- Pin 6 → GND
- Pin 11 → GPIO 17 for the IR receiver
- Pin 12 → GPIO 18 for the IR transmitter

Q1 transistor allows the IR LED to be powered at 5 V DC, increasing its emitting power and providing a limited current on the Raspberry Pi's GPIO, which output 3.3 V DC. The transmitter used is a TSAL7600 by Vishay Semiconductors, a high radiating power 940 nm IR diode in a clear plastic case. If you have some old remote control available, you can also try using its IR LED.

The receiver is an IR38DM (also available as MIM-5385K2 by Unity Opto Technology), a very sensitive IR sensor with preamplifier, demodulator, and filter, calibrated to the frequency of 38 kHz (600 µs burst/pause) with built-in amplifier/squarer (**Figure 6**).

The approach used is to store on Raspberry Pi the signal decoded by the receiver, and then send it back out with the transmitter, matching each gesture with an output signal.

The decoding done by the sensor and an additional library installed on the board will allow us to have a sequence of numbers, which we will store in simple text files, for each remote control button we would like to implement.

Raspberry Pi has native functionality to enable the transmission and reception of IR signal files. We open for editing the file */boot/firmware/conFiguratxt*.

For those unfamiliar with RaspberryOS, we recommend that all file editing instructions be done with the *nano* system editor. This, unlike GUI editors, when opened from the terminal with administrator rights (`sudo bash`) allows opening, editing and saving all types of files, including system files. So from the terminal type:

```
nano /boot/firmware/conFiguratxt
```

(pay attention to the correct patch)

Remove the comments (again as in Python the `#` character) to enable the following lines:

```
# Additional overlays and parameters are documented:
dtoverlay=gpio-ir,gpio_pin=17
dtoverlay=gpio-ir-tx,gpio_pin=18
```

On some versions of RaspberryOS you might not find these lines at all; in this case you simply have to write them yourself. Doing this way, we are asking the operating system to set pin 12 (GPIO18) as the IR transmitter and pin 11 (GPIO17) as the IR receiver.

This operation is possible because these features are native in the Linux kernel. Reboot to make the configurations effective. Now we need to install a key feature to implement our system.

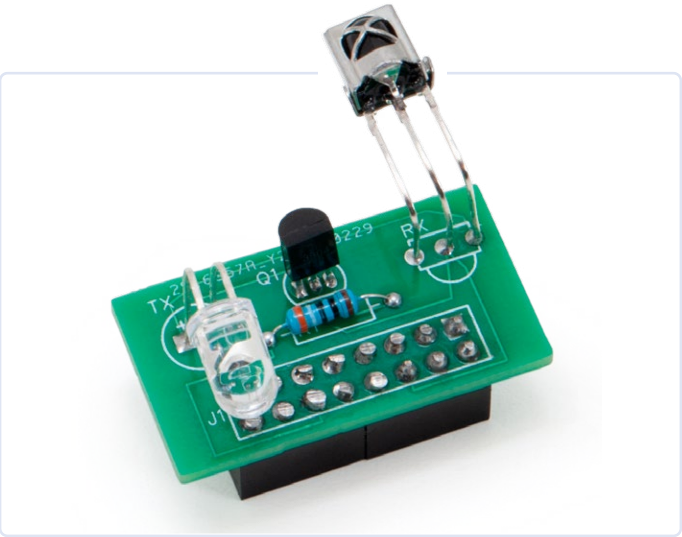
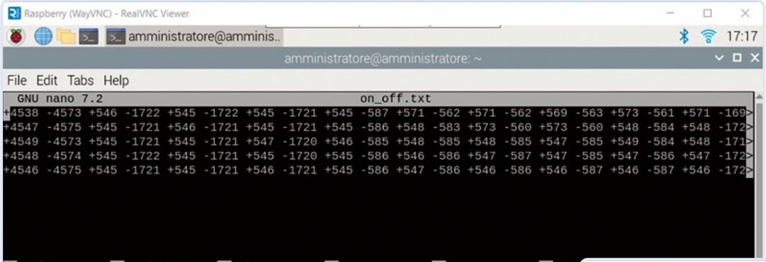


Figure 6: The PCB with TX LED, boost transistor and RX module on the right.

Listing 1: The Python code with the loop for the gesture recognition.

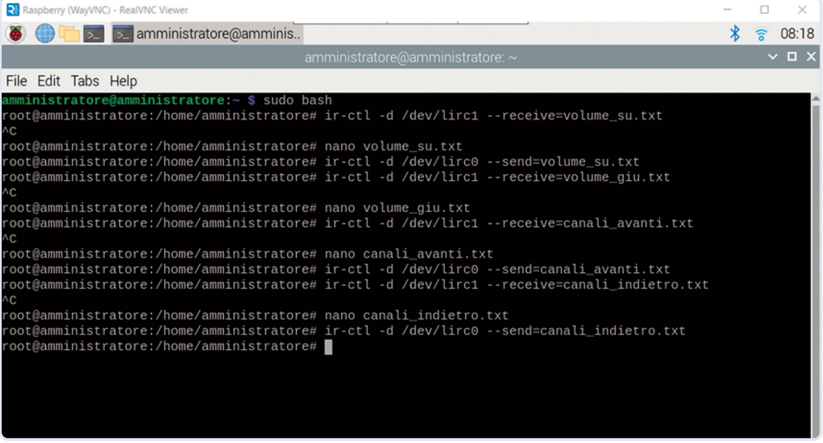
```
# Import of the necessary libraries
import os
import cv2
import mediapipe as mp
from mediapipe.tasks import python
# Maximum number of hands to be detected and path of gesture recognition model
numero_mani = 1
model_path = "gesture_recognizer.task"
# MediaPipe library inizialization
GestureRecognizer = mp.tasks.vision.GestureRecognizer
GestureRecognizerOptions = mp.tasks.vision.GestureRecognizerOptions
GestureRecognizerResult = mp.tasks.vision.GestureRecognizerResult
VisionRunningMode = mp.tasks.vision.RunningMode

# Variable to store the last gesture recognized
last_gesture = None
# Function to print the result of gesture recognition
def print_result(result: GestureRecognizerResult, output_image: mp.Image, timestamp_ms: int):
     global last_gesture

     for single_hand_gesture_data in result.gestures:
         gesture_name = single_hand_gesture_data[0].category_name
         # Checks if the recognized gesture is different from the last detected gesture
         # Used to give an ir inpulse after each gesture
         if gesture_name != last_gesture:
             last_gesture = gesture_name
             # Prints the new recognized gesture
             print("Nuovo gesto riconosciuto:", gesture_name)
             # Correspondences between the recognized gesture and the actions to be performed
             # When you have ir commands available uncomment the os.system instruction
             match gesture_name:
                 case "Open_Palm":
                     print("palmo aperto")
                     # os.system('ir-ctl -d /dev/lirc0 --send=on_off.txt')
                 case "Thumb_Up":
                     print("pollice su")
                     # os.system('ir-ctl -d /dev/lirc0 --send=volume_su.txt')
                 case "Thumb_Down":
                     print("pollice giu")
                     # os.system('ir-ctl -d /dev/lirc0 --send=volume_giu.txt')
                 case "Pointing_Up":
                     print("dito in su")
                     # os.system('ir-ctl -d /dev/lirc0 --send=canali_avanti.txt')
                 case "Victory":
                     print("vittoria")
                     # os.system('ir-ctl -d /dev/lirc0 --send=canali_indietro.txt')
# Options for the gesture recognizer
options = GestureRecognizerOptions(
     base_options=python.BaseOptions(model_asset_path=model_path),
     running_mode=VisionRunningMode.LIVE_STREAM,
     num_hands = numero_mani,
     result_callback=print_result)
```

```
# Creating the gesture recognizer
recognizer = GestureRecognizer.create_from_options(options)
timestamp = 0
mp_drawing = mp.solutions.drawing_utils
mp_hands = mp.solutions.hands
# Initialization of the hand recognizer
hands = mp_hands.Hands(
    static_image_mode=False,
    max_num_hands=numero_mani,
    min_detection_confidence=0.65,
     min_tracking_confidence=0.65)
# Initializing webcam video capture
# If you have multiple video input devices connected to Raspberry
# you need to type the correct number in place of 0
cap = cv2.VideoCapture(0)
# Main program cycle
while True:
     # Capturing a frame by the webcam
     ret, frame = cap.read()
     if not ret:
         break

     # Frame horizontal flip
     frame = cv2.flip(frame, 1)
     # Frame conversion BGR to RGB (MediaPipe compatible)
     frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
     # Frame processing for hand detection
     results = hands.process(frame)
     # Frame conversion from RGB to BGR for visualization
     frame = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)
     # Creating a numpy array with the frame converted to RGB
     np_array = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

     # If hands are detected in the frame
     if results.multi_hand_landmarks:
         for hand_landmarks in results.multi_hand_landmarks:
             # Drawing of hand landmarks on the frame
             mp_drawing.draw_landmarks(frame, hand_landmarks, mp_hands.HAND_CONNECTIONS)
             # Creation of an Image type object compatible with MediaPipe

             mp_image = mp.Image(image_format=mp.ImageFormat.SRGB, data=np_array)
             # Asynchronous gesture recognition on the hand
             recognizer.recognize_async(mp_image, timestamp)
             timestamp = timestamp + 1  # Must monotonously increase in LIVE_STREAM mode

     # Shows the frame with the detected gestures
     cv2.imshow('Riconoscimento Gestuale’, frame)

     # If the Esc key is pressed, stop the cycle
     if cv2.waitKey(1) & 0xFF == 27:
         break
# Release the video capture and close all windows
cap.release()
cv2.destroyAllWindows()
```


Raspberry (WayVNC) - RealVNC Viewer
amministratore@amminis..
17:17
amministratore@amministratore: ~
File Edit Tabs Help

```
  GNU nano 7.2                                   on_off.txt
+4538 -4573 +546 -1722 +545 -1722 +545 -1721 +545 -587 +571 -562 +571 -562 +569 -563 +573 -561 +571 -169>
+4547 -4575 +545 -1721 +546 -1721 +545 -1721 +545 -586 +548 -583 +573 -560 +573 -560 +548 -584 +548 -172>
+4549 -4573 +545 -1721 +545 -1721 +547 -1720 +546 -585 +548 -585 +548 -585 +547 -585 +549 -584 +548 -171>
+4548 -4574 +545 -1722 +545 -1721 +545 -1720 +545 -586 +546 -586 +547 -587 +547 -585 +547 -586 +547 -172>
+4546 -4575 +545 -1721 +545 -1721 +546 -1721 +545 -586 +547 -586 +546 -586 +546 -587 +546 -587 +546 -172>



^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Loc
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go
```

Figure 7: Received signal from remote control, related to on-off button in mode2 encoding. (Source: RealVNC www.realvnc.com/en/)


Raspberry (WayVNC) - RealVNC Viewer
amministratore@amminis..
08:18
amministratore@amministratore: ~
File Edit Tabs Help

```
amministratore@amministratore:~ $ sudo bash
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc1 --receive=volume_su.txt
^C
root@amministratore:/home/amministratore# nano volume_su.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc0 --send=volume_su.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc1 --receive=volume_giu.txt
^C
root@amministratore:/home/amministratore# nano volume_giu.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc1 --receive=canali_avanti.txt
^C
root@amministratore:/home/amministratore# nano canali_avanti.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc0 --send=canali_avanti.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc1 --receive=canali_indietro.txt
^C
root@amministratore:/home/amministratore# nano canali_indietro.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc0 --send=canali_indietro.txt
root@amministratore:/home/amministratore# 
```

Figure 8: Recording and sending the other IR signals from the remote control. (Source: RealVNC www.realvnc.com/en/)

From the terminal:

```
apt install v4l-utils
```

The *v4l-utils* is an open-source utilities package designed to work with Linux video devices (V4L, Video for Linux). V4L is a Linux kernel subsystem that allows applications to interact with video devices, such as webcams, TV tuners, video capture cards and, of course, IR signals.

The *v4l-utils* utilities provide a set of tools to configure, control, and manage these devices. In Listing 1, you noticed that the `ir-ctl` instruction is associated with the `os.system` command (i.e., a Python instruction that directly executes a feature belonging to the system).

It is one of the utilities within *v4l-utils* that allows you to control IR devices. With `ir-ctl`, you can send and receive infrared signals. In the */dev/* directory are *lirc0* file for sending signals and *lirc1* file for transmitting, which are features derived from the LIRC package, now integrated into the OS kernel in the latest versions of Linux. To learn more about, you may pop by here [6].

Put it to the test now, by writing:

```
ir-ctl -d /dev/lirc1 --receive=on_off.txt
```

The terminal will remain waiting for events. Take the remote control you want to "clone", press the *on_off* button while directing it to the IR receiver. Try to press it quickly, and only once, so as not to send too many codes at the same time. Press *ctrl+c* to end the acquisition. Analyze with nano the *on_off.txt* file created.

The file should appear as in **Figure 7**. The received number sequence is in *mode2* which is a specific type of encoding of the captured IR signals. Each pair of numbers represents the pulse lengths (+, or pulse) and the intervals between pulses (-, or space). Mode2 is also a feature of *lirc*, as already mentioned now integrated into the OS. To know more, you may visit [7].

If you have pressed the *on_off* key for too long, you may have stored multiple lines of mode2 codes. You will then have to delete those that repeat. In fact, the first line of the encoding is enough; the others are practically repetitions of the received encoding (actually, you can see the similarity between the lines).

After deleting the unnecessary lines, go to the end of the first one and delete the last value, which represents the timeout between signal sendings (between lines). This value is not needed in the transmission, and it is also easy to identify because it's quite a large number, compared to the others.

Save the file and return to the terminal (*ctrl+x* ; *yes*). Direct the transmitter to the TV and write from the terminal:

```
ir-ctl -d /dev/lirc0 --send=on_off.txt
```

If everything worked, your TV will have turned on "by magic" (or off, if it was already on). Perform the same procedure for the other keys, as illustrated in **Figure 8**. You can, of course, change the matches as you wish. Those for our project are shown in **Table 1**.

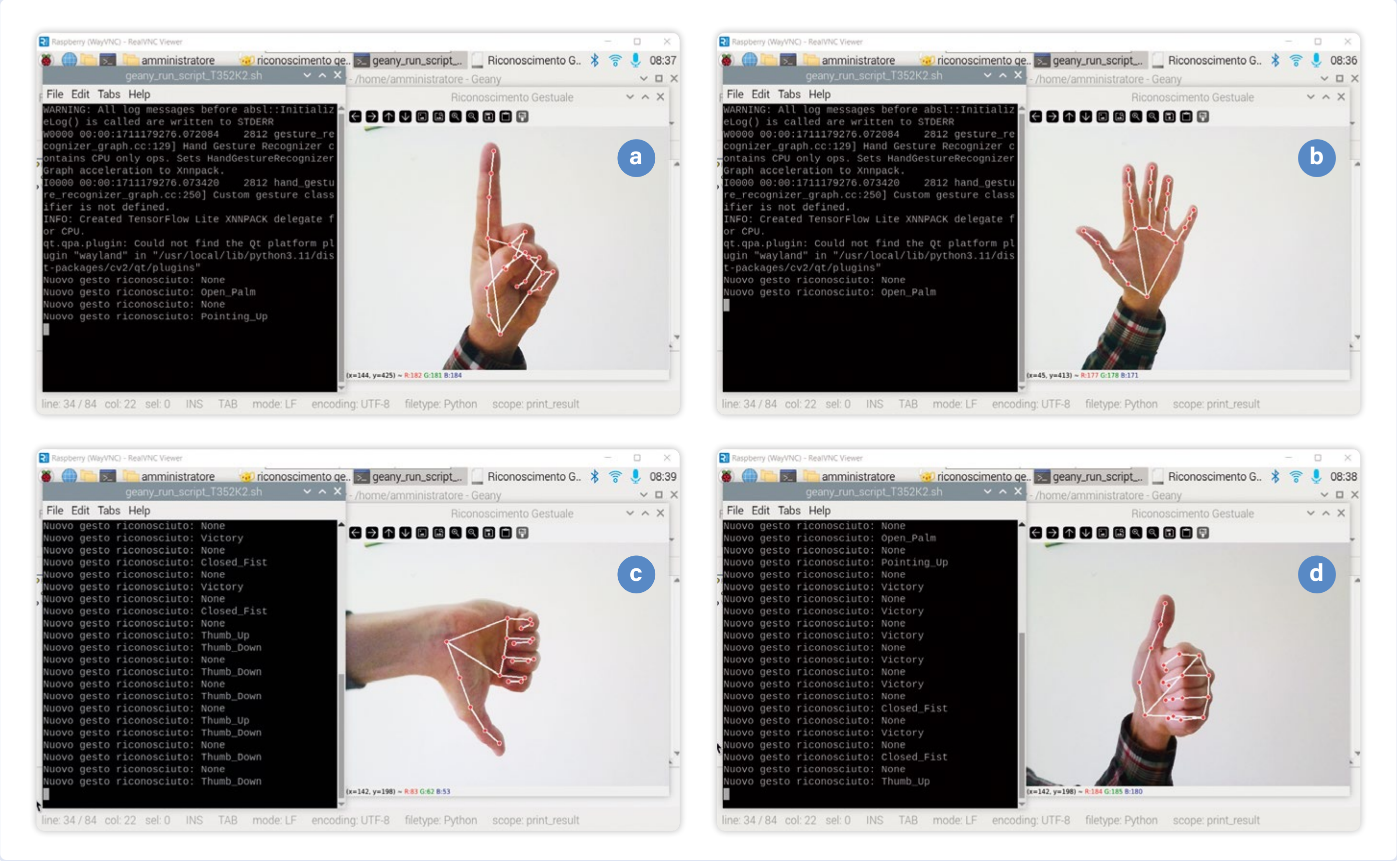
We recommend placing the Python file, MediaPipe template *gesture_recognizer.task* and *txt* files in the main *home* folder (in our project, */home/admin*).

Run the Python file, and place your hand in the webcam frame. Alternate the gesture commands with the fist — which has no matched instructions — so that you can discriminate them properly. **Figures 9a** to **9d** show the system at work, recognizing the finger pointing up (9a), open palm (9b), thumb down (9c) and thumb up (9d).

While you were controlling your device by opening your palm, didn't you feel like a Jedi, at least for a while?

Table 1: Hand gesture correspondences with TV commands.

Hand Gesture	TV Command	txt File
Palm Open	On-Off	on_off.txt
Thumb Up	Volume Up	volume_su.txt
Thumb Down	Volume Down	volume_giu.txt
Finger Pointing Up	Channel Fwd	canali_avanti.txt
Victory Sign	Channel Rev	canali_indietro.txt


a
b
c
d
Raspberry (WayVNC) - RealVNC Viewer
amministratore
riconoscimento ge..
geany_run_script...
Riconoscimento G..
geany_run_script_T352K2.sh
File Edit Tabs Help
Riconoscimento Gestuale
line: 34 / 84 col: 22 sel: 0 INS TAB mode: LF encoding: UTF-8 filetype: Python scope: print_result

(a)

```
WARNING: All log messages before absl::Initializ
eLog() is called are written to STDERR
W0000 00:00:1711179276.072084    2812 gesture_re
cognizer_graph.cc:129] Hand Gesture Recognizer c
ontains CPU only ops. Sets HandGestureRecognizer
Graph acceleration to Xnnpack.
I0000 00:00:1711179276.073420    2812 hand_gestu
re_recognizer_graph.cc:250] Custom gesture class
ifier is not defined.
INFO: Created TensorFlow Lite XNNPACK delegate f
or CPU.
qt.qpa.plugin: Could not find the Qt platform pl
ugin "wayland" in "/usr/local/lib/python3.11/dis
t-packages/cv2/qt/plugins"
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Open_Palm
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Pointing_Up
```

(b)

```
WARNING: All log messages before absl::Initializ
eLog() is called are written to STDERR
W0000 00:00:1711179276.072084    2812 gesture_re
cognizer_graph.cc:129] Hand Gesture Recognizer c
ontains CPU only ops. Sets HandGestureRecognizer
Graph acceleration to Xnnpack.
I0000 00:00:1711179276.073420    2812 hand_gestu
re_recognizer_graph.cc:250] Custom gesture class
ifier is not defined.
INFO: Created TensorFlow Lite XNNPACK delegate f
or CPU.
qt.qpa.plugin: Could not find the Qt platform pl
ugin "wayland" in "/usr/local/lib/python3.11/dis
t-packages/cv2/qt/plugins"
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Open_Palm
```

(c)

```
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Victory
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Closed_Fist
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Victory
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Closed_Fist
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Thumb_Up
Nuovo gesto riconosciuto: Thumb_Down
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Thumb_Down
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Thumb_Down
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Thumb_Up
Nuovo gesto riconosciuto: Thumb_Down
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Thumb_Down
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Thumb_Down
```

(d)

```
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Open_Palm
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Pointing_Up
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Victory
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Victory
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Victory
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Victory
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Victory
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Closed_Fist
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Victory
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Closed_Fist
Nuovo gesto riconosciuto: None
Nuovo gesto riconosciuto: Thumb_Up
```

Figure 9: These four images show the main gestures recognized by the system. On the right of each one, you may read the recognition sequences, that vary according to the outcome of the process. (Source: RealVNC www.realvnc.com/en/)

Further Applications

We have begun to explore the world of artificial intelligence and machine learning through a simple project, which nonetheless can enable the construction of a course of study in a new and fascinating field, towards which modern technology is rapidly heading. Also in the maker field, Raspberry Pi offers the possibility of replacing our PCB with any output device, allowing us to control, for example, any kind of electrical devices in our home.

A downloadable set of files for SW and PCB layouts for this project is available at the Elektor Labs webpage [8]. ◀

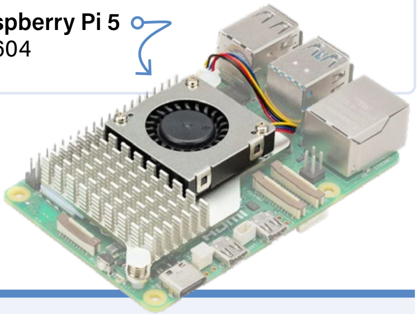
240433-01

Questions or Comments?

Do you have technical questions or comments about this article? You may write to the editorial team of Elektor at editor@elektor.com.

Related Products

- **Raspberry Pi 5 (8 GB RAM)**
  www.elektor.com/20599
- **Active Cooler for Raspberry Pi 5**
  www.elektor.com/20604



WEB LINKS

[1] European Safety Transport Council on EuroNCAP: https://tinyurl.com/3rs8z9cv
[2] MediaPipe Studio gesture recognition webpage: https://tinyurl.com/44jxnep8
[3] MediaPipe customization guide: https://tinyurl.com/y9bbpu5h
[4] RealVNC VNC viewer download page: https://tinyurl.com/3dvhjfru
[5] MediaPipe gesture recognition task download: https://tinyurl.com/4xsrfz8z
[6] LIRC webpage: https://lirc.org/
[7] Ubuntu webpage for ir-ctl tool usage: https://tinyurl.com/68xpjd22
[8] Elektor Labs webpage for this article: https://www.elektormagazine.com/labs/ai-based-universal-ir-remote-control

AI for Product Concept Design

A Tour of the World of AI-Based Art Generators

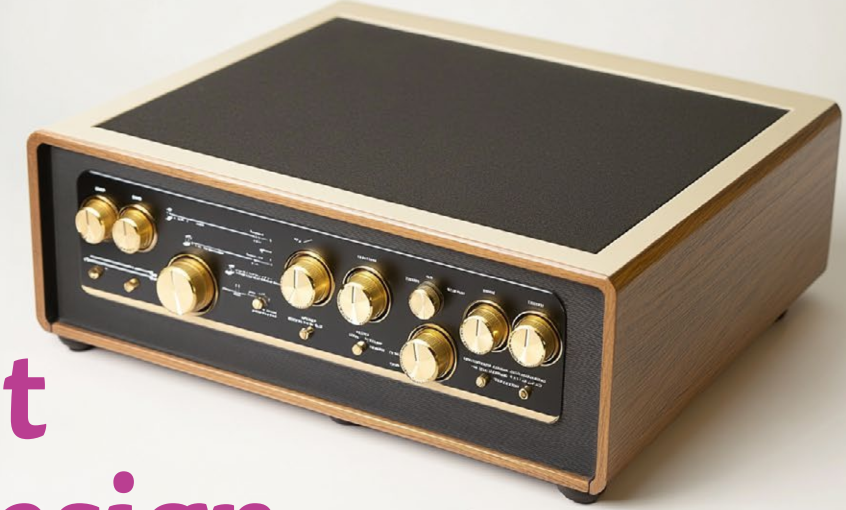


Figure 1: Rendering of a Hi-Fi amplifier created with Midjourney by the author.

By Edwin van den Oetelaar (The Netherlands)

Over the past year, AI for image creation has taken a huge leap, making this technology accessible to everyone. Does this give electronics hobbyists and creators a powerful new tool? A few sketches and a well-formulated text prompt are all it takes to get a photorealistic representation of our ideas. But is it really that simple? And what use is it, in practice?

Since the advent of AI, I have always been truly interested in its creative capabilities, and immediately wanted to try the new tools made available on the Web, such as Midjourney, Stable Diffusion, Freepik Sketch, and Dall-E 3.0. In this article, I would like to give you an opinion on them, although this list is not exhaustive and at the time of publication may be already out of date. In my experiments, I tried to create a design concept for an audio amplifier that might look beautiful. Using a combination of sketches, photos of existing devices and directed prompts, I tried to bring my vision to life using Midjourney and other tools. Along the way, I'll explain jargon terms such as *prompt-engineering*, *inpainting*, and *style-transfer*.

Moving at Lightning Speed

If I were to start from an initial observation, I would have to say that developments in Gen-AI image generation are moving at lightning speed. New online services and models appear every week. Image quality is improving by leaps and bounds. There now seems to be a product for everything.

Convenience serves man — if you can't come up with a prompt yourself or aren't creative, there is a paid service that will help you with sample prompts, styles and specific templates for every challenge, such as logo design, color palettes, app designs and fashion design. Everyone seems to be jumping on this train, and commercial parties are integrating AI into their existing services. In addition to these online services, more and better offline models and tools are becoming available, such as Flux-1-Schnell and Stable Diffusion mentioned above.

Many photorealistic portraits created with these generators are barely distinguishable from real photographs, even to practiced eyes. Effects that were difficult to realize just a few weeks ago — such as text in images — are now no longer a problem. Distorted limbs became rare.

AI offers a new source of inspiration. With simple textual input, AI can generate different variations of a design, which can lead to unexpected ideas. Tools such as Midjourney and Stable Diffusion XL allow users to create multiple mockups from a sketch or prompt, which helps discover design options that might otherwise be overlooked.

Figure 1 shows a rendering of a Hi-Fi amplifier created with Midjourney. (Refer to text box **Get Started with Midjourney**.)

Get Started with Midjourney
How to get started with *Midjourney*: a one-click roadmap!

- Go to Midjourney's official website [1].
- Click "Sign In" to log in with your Discord account or Google account.
- That's all.

Prompt Engineering

Prompt engineering is the targeted formulation of text commands to guide Gen-AI, such as Midjourney and Stable Diffusion, in creating images. In the context of product design, this means using precise words and descriptions to direct the AI to generate exactly the desired shapes, styles and details. It's all about precision in your task, so that the Gen-AI creates images that match your vision. This can be tricky, since the same prompt always yields different variants, as shown in the four examples on **Figure 2**. This is because AI uses a random number, called *seed*, which you can also fix with Stable Diffusion.

A common use of *ChatGPT* (also a Gen-AI) is to generate prompts. You can ask ChatGPT how best to set up a prompt, such as for Midjourney. You will then get an explanation of how such a prompt is constructed, weighting factors, flags, etc., with which you can further experiment and iterate.

The Ease of Editing and Refining

A major advantage of Gen-AI is that modifying or deleting existing ideas is much easier than starting from scratch. This iterative process allows designers to quickly explore different options. This fits well with the modus operandi of many young hobbyists: try something out first, then tweak it, where the quick result is a thrill. Selectively removing elements from a generated image and then having it re-generated is called *selective inpainting*. **Figure 3** shows an example of this technique.

Limitations of Fun Graphics

Whether it is Gen-AI, 3D printing or laser cutting, there are often countless iterations without regard to cost. It's fun and fast, but without a clear vision of the design, and without the right dimensions and materials to match the function, you'll get nowhere. Thinking through what you want in advance, thinking through the use cases and planning, is essential to working efficiently and purposefully. You can't give engineering parameters without design-rules, materials, dimensions, without respecting any criteria or logic.

The Addiction Factor, Keep Varying!

When you see what people generate with Midjourney, it seems like it's more of a game than a tool. People keep endlessly creating new variations, hoping that a masterpiece will spontaneously emerge. You can spend an entire evening trying out styles of photographers and artists like this. For hobbyists without time constraints, such as those in their attic, the endless generation of iterations and variations can be addictive, just like other online activities. Each time you view a new variation, you get a dopamine kick, similar to what you experience when gaming or gambling.

Limitations, Mediocrity

Gen-AI such as Midjourney and Stable Diffusion work from a large data set of existing images. AI combines elements from this data to create a kind of "average" view. This can make it feel that truly new images are difficult to create, since Gen-AI always falls back on what it has "seen" before.

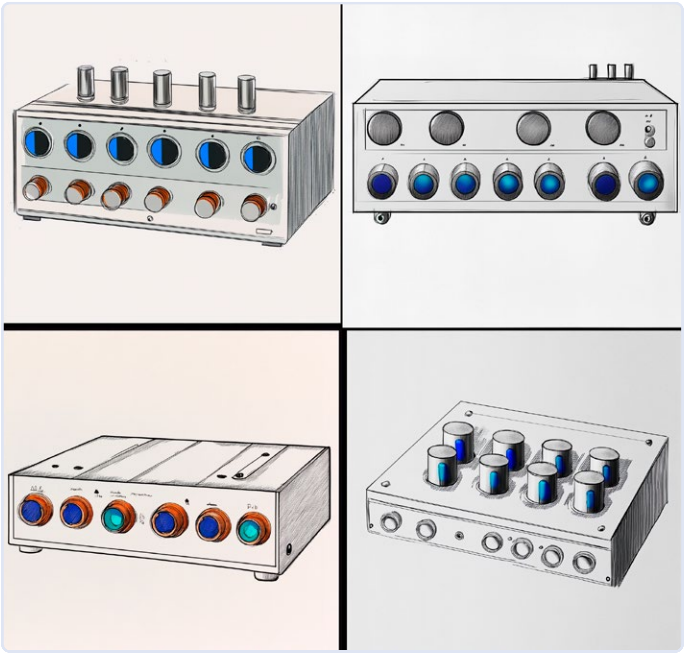
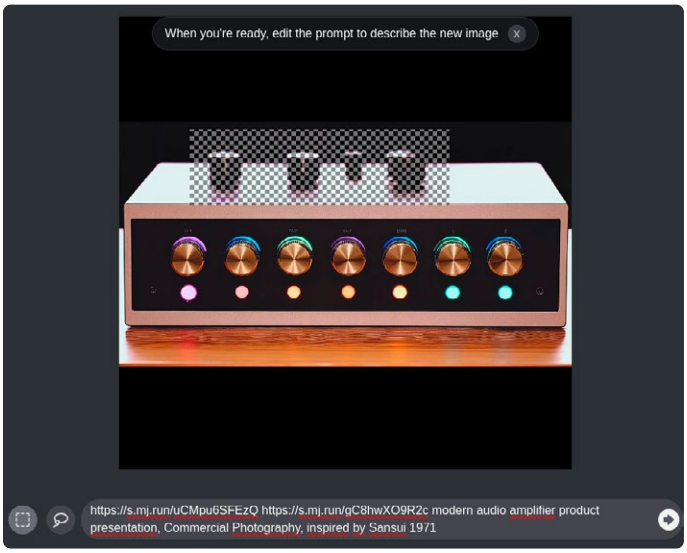


Figure 2: The same prompt produced four different variants of amplifier.



When you're ready, edit the prompt to describe the new image

https://s.mj.run/uCMpu6SFEzQ https://s.mj.run/gC8hwXO9R2c modern audio amplifier product presentation, Commercial Photography, inspired by Sansui 1971

Figure 3: Application of the selective inpainting technique, to remove an unwanted part from the generated image.

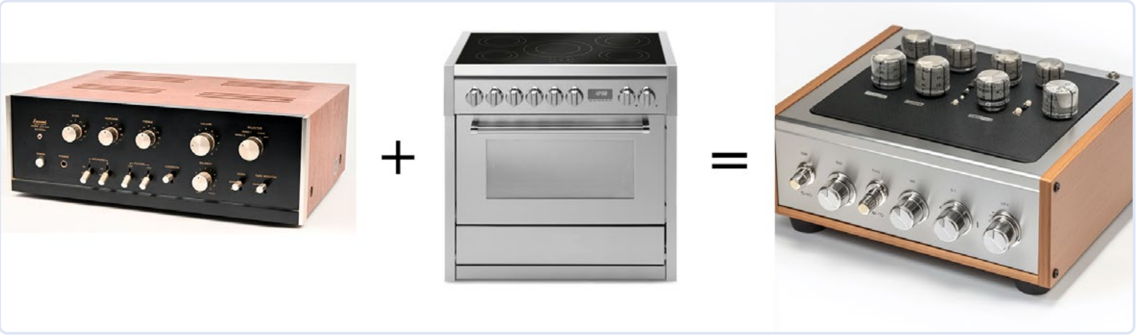


Figure 4: Example of AI-generated image that comes from the blending and the interpretation of the two source images.

Smart prompt engineering allows users to force unique and surprising combinations. The power of AI lies precisely in combining elements in unexpected ways, creating images that feel original and different, even when based on existing data. While AI cannot create entirely new things from scratch, it can produce results that are perceived as unique because of the way it reinterprets familiar elements.

Using Your Own Images

Incorporating your own photos and images into the AI design process can significantly improve results. By uploading images of existing designs or inspiring objects, the Gen-AI can generate more specific and relevant designs. In Midjourney, you can upload and merge multiple images via the `/blend` function and also combine them with a prompt, as shown on **Figure 4**, which illustrates the merge of two images into a final one.

Use of Weighting Factors and Style References in Prompts

Weighting factors in prompts can be used to emphasize certain elements of the design. This provides more control over the result and helps create designs that more accurately reflect the intended vision. You can use `::` and numeric values for this purpose.

One example in Midjourney: If you want the AI to emphasize a device in the foreground and not in the background, you can do this with a prompt such as:

```
/imagine a high-tech speaker on a table in a modern living room, clean design, minimalist ::2 background blurred ::0.5
```

Here `minimalist` has a weighting factor of 2 compared to the other elements and `background blurred` just a weighting factor of 0.5.

Style references refer to another image to adopt a specific style. You do this by using `--sref` followed by the URL of the reference image.

In Midjourney, if you want to adopt a specific style of an image, use, for example:

```
/imagine futuristic device with blue accents --sref https://example.com/image.jpg
```

Cost and Accessibility

Traditional product design can be time-consuming and expensive, especially for hobbyists working with limited resources. Gen-AI tools significantly lower this barrier by making it possible to generate designs quickly and inexpensively. Instead of spending hours sketching and modeling, designers can now create dozens of images within minutes.

The availability of affordable or free Gen-AI makes this technology accessible to a wide audience. Tools like OpenAI's DALL-E (via ChatGPT website) and Flux or Stable Diffusion (via *poe.com*) offer even free access to basic functions, allowing hobbyists to get started in no time.

For makers, it may not be worth learning all the ins and outs of GenAI tools. These tools change quickly and become obsolete at the same pace. In addition, creating graphics is often not up the alley of electronics engineers, who are more focused on function than form. Look around you for people whose hobby is GenAI-prompting, just as you have your own hobby. Collaborating with them can be not only enjoyable, but also mutually instructive.

Overview of Online AI Art Generators

Bing Image Creator

- AI art model: DALL-E 3.
- Platform: Web.
- Cost: Free of charge.

Bing Image Creator is the result of the collaboration between Microsoft and OpenAI. It uses the latest DALL-E model, DALL-E 3, and is currently available for free.

DALL-E 3 (via ChatGPT)

- Platform: Web (via ChatGPT).
- Cost: $20/month as part of ChatGPT Plus.

DALL-E 3 is a substantial upgrade over DALL-E 2, especially if you have a ChatGPT Plus subscription. The results are significantly better, and you have more control by combining it with ChatGPT.

Canva

- AI art model: Stable Diffusion.

- Platform: Web, iOS, Android.
- Cost: Free; from $12.99/month for Pro with more AI features.

Canva recently added a text-to-image art generator. This feature integrates seamlessly with the rest of the template-based design app, allowing you to add AI-generated art to everything from social media posts to birthday cards.

NightCafe

- AI art models: Stable Diffusion, DALL-E 2, CLIP-Guided Diffusion, VQGAN-CLIP.
- Platform: Web.
- Cost: From $6/month for 100 credits (good for about 1,240 images per month).

NightCafe offers additional features such as styles for DALL-E 2 and Stable Diffusion, and allows you to use older generative art models.

OpenArt

- AI art models: Stable Diffusion, DALL-E 2, and other open-source models.
- Platform: Web.
- Cost: Free for 50 trial credits; from $10/month for 5,000 credits.

Like NightCafe, OpenArt offers additional features for Stable Diffusion and DALL-E 2, plus access to additional open-source models. You have more control over the specific details of the images you generate. OpenArt also offers options such as sketch-to-image and a stock art transformer that adapts stock images to your needs.

Adobe FireflyFl

- Platform: Web, Adobe Express, Adobe Photoshop, and other Adobe tools.
- Cost: Free for 25 credits per month; from $5/month for 100 credits per month (and included with various Adobe subscriptions).

Although you can use it online, it is now integrated directly into Adobe products such as Express and Photoshop.

One of Firefly's best features is the ability to create custom text effects with a written prompt.

Quora Poe

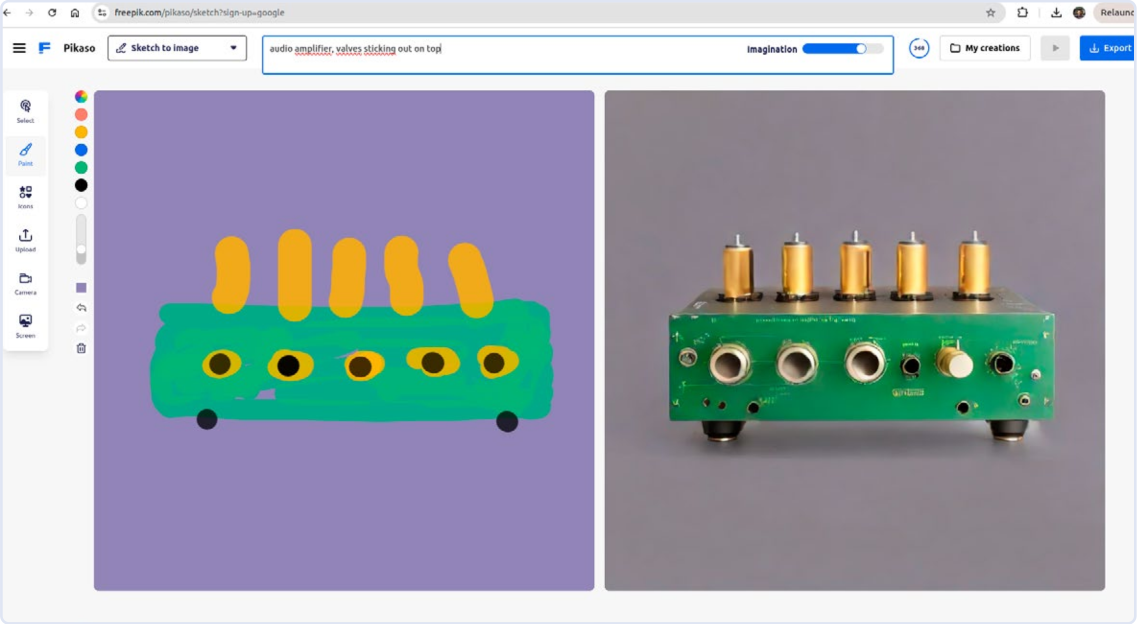
A platform, Quora Poe, an application or service created by Quora, is available on the Web. This platform allows users to interact with various AI models, including ChatGPT and other AI-based chatbots. On it you can find, among others, FLUX-schnell at [2] and Stable Diffusion at [3]. There will probably be a limited number of credits, but you can try them out.

Other Online Services

In my overall tinkering, I've also experimented with FreePik [4]. This tool offers an interesting feature: in addition to entering prompts — like Midjourney or Stable Diffusion — you can sketch in real time. As you draw, the image adapts to your sketch.

What is noticeable, however, is that the image is constantly changing, making it difficult to control accurately. Sizing and technical precision are lacking in all ImageGen-AI. **Figure 5** shows an example of rendering: the rendered image on the right is derived directly from the (very) rough sketch made by hand on the left.

FreePik is free for a limited number of actions per day, but if you really want to get started, a subscription is required. In addition to FreePik, there are numerous new online services now available.



freepik.com/pikaso/sketch?sign-up=google

audio amplifier, valves sticking out on top

Figure 5: AI helps! The rendered image on the right has been created by Freepik starting from the (very) draft, hand-made sketch on the left.

Of course, there are many others on the web: just type: “AI art generators” in your search bar, and you’ll discover a world that you probably weren’t aware of!

Offline Alternatives

Flux.1-Schnell (I recommend this one): A fast and efficient implementation of Stable Diffusion, designed to run on consumer hardware.

- PROs: Fast, efficient, and can run on less powerful hardware.
- CONs: Possibly less customizable than the full Stable Diffusion implementation.

To test Flux.1 online or run it at home, visit [5]. Keep in mind that you will need a powerful video card if you want to install yourself, preferably an Nvidia with at least 12 GB VRAM and many CUDA cores. The more powerful the card, the less latency you will experience.

Stable Diffusion web UI: if you would like to get started with this one on your own PC, you may take a look at [6]. Again, a powerful video card is essential for a smooth operation.

Technical Limitations of Gen-AI in Design

Technical constraints on form and function make the engineer’s job both interesting and indispensable. An engineer must consider a wide range of requirements-from functional performance to manufacturability, maintainability and recyclability, also known as *Design for X* (DFx). Gen-AI tools are currently not (yet) able to provide for this process; while they can quickly generate pretty pictures, the real value of good design lies in the practical application of engineering knowledge and experience.

Future of Gen-AI in Design

New tools are becoming available daily; it feels like an adventure, new discoveries every day. However, the role of the engineer remains crucial in creating functional and manufacturable designs. Gen-AI can currently serve primarily as a tool for quickly visualizing ideas, but the actual design process requires deep collaboration between the engineer, designer and industrial designer to meet all the complex requirements. Today’s AI tools do not yet sufficiently help in this regard, but they can contribute to the creative process by quickly generating visually appealing concepts. So for now, their usefulness is limited to this aspect of the design process.

230514-01

About the Author

Ing. Edwin van den Oetelaar is an expert and tech coach at Fontys ICT. He works with students on cool projects like building robots, self-driving cars, and other electronics. Edwin loves teaching by doing, helping students learn real skills through hands-on projects. Whether it’s designing circuits or creating new tech, he enjoys exploring the latest technologies and sharing his knowledge. In his leisure time, he likes experimenting with electronics and keeping up with industry trends. For him, learning and creating should be fun, and he loves helping the next generation of engineers to experience that too.

Questions or Comments?

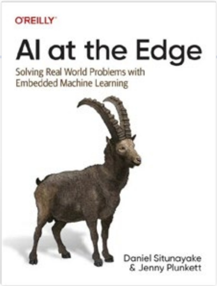
Do you have technical questions or comments about this article? You may write to the editorial team of Elektor at editor@elektor.com.

Author's Note

Those interested in learning more about AI may read the book, *Living and Learning With AI*, written (in Dutch) by my ICT faculty colleagues [7].

Related Product

- **D. Situnayake, Jenny Plunkett, *AI at the Edge* (O’Reilly, 2023)**
  www.elektor.com/20465


O'REILLY
AI at the Edge
Solving Real World Problems with Embedded Machine Learning
Daniel Situnayake & Jenny Plunkett

WEB LINKS

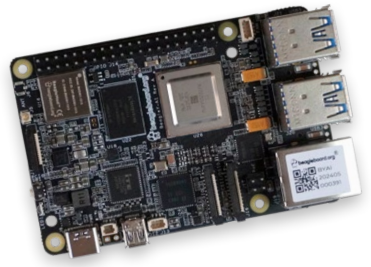
[1] Midjourney’s official website: https://www.midjourney.com/
[2] FLUX-schnell at Quora Poe website: https://poe.com/FLUX-schnell
[3] Stable Diffusion XL at Quora Poe website: https://poe.com/StableDiffusionXL
[4] Pikaso’s website: https://www.freepik.com/pikaso
[5] Website for Flux.1 testing: https://huggingface.co/black-forest-labs/FLUX.1-schnell
[6] Stable Diffusion on GitHub: https://github.com/AUTOMATIC1111/stable-diffusion-webui
[7] R. Huijts, K. Suilen, D. Bloks, E. Saçan ”Leven en Leren met AI” (Noordhoff Business): https://tinyurl.com/hdej6mpm

Get your hands on cutting-edge AI Hardware!

Raspberry Pi AI Kit

The Raspberry Pi AI Kit includes the Raspberry Pi M.2 HAT+ and a Hailo AI acceleration module for use with the Raspberry Pi 5. It provides an accessible, cost-effective, and power-efficient way to integrate high-performance AI. Explore applications including process control, security, home automation, and robotics!

www.elektor.com/20879



BeagleY-AI SBC with GPU, DSP and AI Accelerators

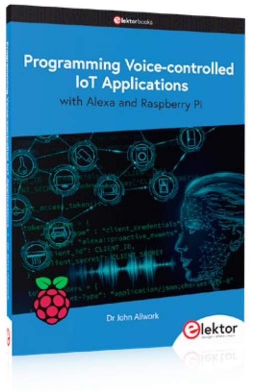
BeagleY-AI is a low-cost, open-source, and powerful 64-bit quad-core single-board computer, equipped with a GPU, DSP, and vision/deep learning accelerators, designed for developers and makers. Users can take advantage of BeagleBoard.org's provided Debian Linux software images, which include a built-in development environment.

www.elektor.com/20991

Raspberry Pi AI Camera

The Raspberry Pi AI Camera is a compact camera module based on the Sony IMX500 Intelligent Vision Sensor. The IMX500 combines a 12 MP CMOS image sensor with on-board inferencing acceleration for various common neural network models, allowing users to develop sophisticated vision-based AI applications without requiring a separate accelerator.

www.elektor.com/20953



Programming Voice-controlled IoT Applications with Alexa and Raspberry Pi

This book is divided into two parts: creating Alexa skills and designing Internet of Things (IoT) devices using a Raspberry Pi. It covers topics like Alexa skill development, in-skill purchasing, and using AWS Lambda, as well as building smart home devices controlled by Alexa. Readers also learn about MQTT messaging, creating proactive notifications, and turning a Raspberry Pi into a stand-alone Alexa device.

www.elektor.com/20400

Unitree Go2 Pro Quadruped Robot

The Unitree Go2 is a quadruped robot designed for the research & development of autonomous systems in the fields of human-robot interaction (HRI), SLAM & transportation. Due to the four legs, as well as the 12DOF, this robot can handle a variety of different terrains.

www.elektor.com/20357

Waveshare Jetson Orin Nano AI Development Kit

This AI Edge Computing Development Kit is based on the Jetson Orin Nano Module providing rich peripheral interfaces such as M.2, DP, USB, etc.

The kit also comes with a pre-installed AW-CB375NF wireless network card that supports Bluetooth 5.0 and dual-band WIFI, with two additional PCB antennas, for providing high-speed and reliable wireless network connection and Bluetooth communication.

www.elektor.com/20762

Join the Elektor Community

elektor MAG
design > share > earn
Audio DSP FX Processor Board
Versatile Platform with ESP32 + ADAU1701 DSP
Focus on Prototyping, Production & Components
PCB Layout and Safety
KiCad 8
THIS EDITION
BONUS CONTENT
Opamp Tester

Take out a GOLD membership!

- ✓ The Elektor web archive from 1974!
- ✓ 8x Elektor Magazine (print)
- ✓ 8x Elektor Magazine (digital)
- ✓ 10% discount in our web shop and exclusive offers
- ✓ Access to more than 5,000 Gerber files

Also available

The Digital GREEN membership!

- ✓ The Elektor web archive from 1974!
- ✓ 8x Elektor Magazine (digital)
- ✓ 10% discount in our web shop and exclusive offers
- ✓ Access to more than 5,000 Gerber files

www.elektormagazine.com/member

elektor
design > share > earn