

## 1. Remove background

A recurring task is to remove the background from a portrait or picture of a person. You can do this by hand, but that takes time.

### 1.1 The Python script

There is a Python script, called `rembg` that removes the background from an image. Now, one of the problems of Python is illustrated by "<https://xkcd.com/1987/>" and this script has quite a number of dependencies. And the dependencies are not installed automatically. Some of the dependencies may interfere with your global Python environment. The solution is to create a virtual environment for this script.

My virtual environment will be under `/home/ljm/src/python_venv` because I thought that was a good place. I `cd -ed` to that directory and created the virtual environment:

```
python3 -m venv rembg
```

Use `python3` because it is the only way that it is guaranteed to work.

Next, activate that environment:

```
rembg/bin/activate
```

Next step is to install the packages. You might expect that dependencies are installed. No. This is Python. So, you should install:

```
pip install rembg
pip install onnxruntime
pip install click
pip install filetype
pip install watchdog
pip install aiohttp
pip install gradio
pip install asyncer
```

Run `rembg` by hand to see if I've missed some dependencies.

And, to see how great Python is,(1) do

```
du -sh rembg
937M  rembg
```

That's almost a gig for a single script to work normally. Isn't Python great?

---

(1) I loath Python

## 1.2 A shell wrapper

To hide the horrors of the virtual Python environment, I created a script.

```
#!/bin/bash
#INSTALL@ /usr/local/bin/rembg
#INSTALLEDFROM verlaine:/home/ljm/src/python_venv

# Path to the virtual environment
# Adapt this to your own environment
VENV_PATH=/home/ljm/src/python_venv/rembg

# Activate the virtual environment
source "$VENV_PATH/bin/activate"

# Run the Python script
"$VENV_PATH/bin/rembg" i "$@"

# Deactivate the virtual environment
deactivate
```

## 1.3 The GIMP plugin

To run it from GIMP, you need to create a GIMP plugin.

```
/*
#INSTALL_C@ ~/.config/GIMP/2.10/plug-ins/plugin_rembg.c
#MAKE gimptool-2.0 --install plugin_rembg.c
*/

#include <libgimp/gimp.h>
#include <glib.h> // For g_strdup_printf, g_spawn_command_line_sync
#include <unistd.h> // For unlink()

// Declare a global variable for the JPEG file path
char *jpegfile;
char *jpegfile2;

static void query(void);
static void run(const gchar *name, gint nparams, const GimpParam *param, gint *nreturn_vals, GimpParam **return_vals);

GimpPlugInInfo PLUG_IN_INFO = {
    NULL, // Init function (optional)
    NULL, // Quit function (optional)
    query, // Query function (required)
    run // Run function (required)
};

// Plugin query function
static void query(void) {
    static GimpParamDef args[] = {
        { GIMP_PDB_INT32, "run_mode", "Run mode" },
        { GIMP_PDB_IMAGE, "image", "Input image" },
        { GIMP_PDB_DRAWABLE, "drawable", "Input drawable" }
    };
};

gimp_install_procedure(
    "plug-in-rembg", // Procedure name
    "Remove background", // Plugin description
    "Saves the current image as a JPEG file, runs the external rembg script, and imports the result as a new layer", // Plugin help
    "ljm", // Author
    "ljm", // Copyright
    "2025", // Date
    "<Image>/Filters/Misc/Remove Background", // Menu path
    "RGB*, GRAY*", // Image types this plugin works with
    GIMP_PLUGIN, // Plugin type
    G_N_ELEMENTS(args), // Number of input parameters
    0, // Number of output parameters
    args, // Input parameter definitions
    NULL // No output parameters
);

// Plugin run function
static void run(const gchar *name, gint nparams, const GimpParam *param, gint *nreturn_vals, GimpParam **return_vals) {
    gint32 image_id, drawable_id, new_layer_id;
    GError *error = NULL;
    gchar *stdout_output = NULL;
```

```

gchar *stderr_output = NULL;
gint exit_status;

// Generate a unique temporary filename in /tmp
jpegfile = g_strdup_printf("/tmp/tempfile-%d.jpg", g_random_int());
jpegfile2 = g_strdup_printf("/tmp/tempfile2-%d.jpg", g_random_int());

// Get the image and drawable ID from the input parameters
image_id = param[1].data.d_int32;
drawable_id = param[2].data.d_int32;

// Save the image to the specified file using gimp_file_save
gboolean success = gimp_file_save(GIMP_RUN_NONINTERACTIVE, image_id, drawable_id, jpegfile, jpegfile);

if (!success) {
    gimp_message("Error: Could not save the image as JPEG.");
    g_free(jpegfile);
    return;
}

// Run the external script with the JPEG files as an argument
gchar *cmd = g_strdup_printf("/usr/local/bin/rembg %s %s", jpegfile, jpegfile2);

// Execute the command and capture stdout and stderr
success = g_spawn_command_line_sync(cmd, &stdout_output, &stderr_output, &exit_status, &error);

// Check if the command ran successfully
if (!success) {
    gimp_message("Error: Could not run the rembg script.");
} else {
    // Display the output from stdout (if any) in the GIMP message bar
    if (stdout_output && *stdout_output) {
        gimp_message(stdout_output);
    }

    // Optionally, handle stderr output
    if (stderr_output && *stderr_output) {
        gimp_message(stderr_output);
    }

    // Load the enhanced JPEG as a new layer
    new_layer_id = gimp_file_load_layer(GIMP_RUN_NONINTERACTIVE, image_id, jpegfile2);

    if (new_layer_id != -1) {
        // Add the new layer to the image
        gimp_image_insert_layer(image_id, new_layer_id, -1, -1);
    } else {
        gimp_message("Error: Could not load the enhanced image as a new layer.");
    }

    // Delete the temporary file after importing the layer
    unlink(jpegfile); // Use standard POSIX unlink to remove the file
    unlink(jpegfile2); // Use standard POSIX unlink to remove the file
}

// Free resources
g_free(jpegfile);
g_free(cmd);
g_free(stdout_output);
g_free(stderr_output);

// No return values to set
*return_vals = 0;
}

// Main entry point
MAIN();

```