**Folder setup – do this as root**

1. Download the files from Github, it is the latest update.
2. Copy all HTML and PHP files into your /var/www/html folder, change all permissions to www-data, and chmod 755.

**cat <file> > /var/www/html/<file> [or mv is fine too]**

**chown www-data:www-data /var/www/html/<file>**

**chmod 755 /var/www/html/<file>**

1. Copy all cgi, python, and bash scripts into /usr/lib/cgi-bin/, also change permissions to www-data and chmod 755. You don’t need to copy the frames/ folder, it should be automatically re-created when running the video processing scripts later.
2. Check that all the permissions are correct.
3. ll the /var/www/html folder. In /var/www/html, you should see a folder called cgi-bin. Check if the cgi-bin folder links to /usr/lib/cgi-bin. If it is, it would look like this when you ll:

cgi-bin -> /usr/lib/cgi-bin

If not, then delete it first. Just do rm -r cgi-bin. Then create the symlink using this command:

ln –s /usr/lib/cgi-bin cgi-bin/

Check again if the symlink is created correctly.

In the /var/www/html folder, there should be a folder called vids/. chmod 755 the vids/ folder and change permissions for group:

chgrp www-data vids/

cd into cgi-bin and delete the folder vids, using rm -r vids/. Then create the symlink:

ln -s ../../../var/www/html/vids

Check again if the symlink exists. If you cd into cgi-bin’s vids, it should redirect you back to /var/www/html’s vids/ folder.

**Install necessary stuff – I’m not sure 100% of the download modules’ names, double-check if they don’t work**

1. Install postgresql – apt-get install psql
2. Install psycopg2 – apt-get install python-psycopg2
3. Install bcrypt – apt-get install bcrypt
4. Download OpenFace and follow instructions from this website (if you haven’t already, I’m sure most of you already have OpenFace installed in your machine, especially Ty, who needed it to run his scripts): <https://github.com/TadasBaltrusaitis/OpenFace/wiki/Unix-Installation>

After all this, I think if you try to upload a file from video\_upload.html it should work and save the video in /var/www/html’s vids/ folder and split the videos into still images, at least. Running the entire script might still give you an error, since there are a few locations I still need to change in your environment.

1. If upload fails, try running <?php phpinfo(); ?> from Apache, and find where your php.ini is located. Open php.ini, and change the default values of memory\_limit, upload\_max\_filesize, post\_max\_size, and I think max\_memory\_size (I forget if this one exists or not). Change them all to something big like 32M. Restart apache (service apache2 restart). Try uploading file again.

Oh, and don’t forget to create the tables using postgresql! Login: sudo –u postgres psql postgres. Change the password for user postgres to ‘postgres’:

ALTER USER postgres WITH PASSWORD ‘postgres’

Then create schema, then create tables. Updated database tables are in Github.

Let me know if this still gives you problems, and I can try to troubleshoot from there. Ways to debug/troubleshoot on your own:

1. Open /var/log/apache2/error.log for errors running stuff with Apache.
2. Try-catch with PHP? try {… } except as Exception e { print str(e) };
3. If upload file still won’t work, try downloading chrome and try it with Chrome. Or, delete all cache and cookies in Mozilla before retrying.
4. If everything fails, try at least to run the video processing scripts using your terminal (as root), and see if it works. Copy a sample video to vids/ folder in /var/www/html, chmod 755-ed and chown-ed to www-data. From /var/www/html, run bash cgi-bin/process\_vid <filename>. See if it runs, or where it fails. Or, if you want, you could run the processing video scripts one by one. Order:

bash cgi-bin/videosplit <filename>

python cgi-bin/videometa.py <filename>

python cgi-bin/getpoints.py <filename> 🡪 in here I call /home/petr/Desktop/OpenFace/FaceLandmarkImg, we should change that to wherever your OpenFace folder is located.

python cgi-bin/makevid.py <filename> 🡪 if this doesn’t work, try running delaunay\_triangle.py first with <filename> <frame number> to see if it’s the delaunay\_triangle.py giving you problems.