

Bitácora de trabajo

Susana Astorga Rodríguez

Viernes 24 setiembre:

- Primera reunión de grupo para leer a detalle el documento del proyecto, comentar dudas y plantear método de trabajo.
- Este día se creó un repositorio de trabajo en github.
- Se investigó acerca del tema de reconocimiento facial.
- Se investigaron las posibles dependencias a necesitar.
- Se realizó el planeamiento de la ejecución.

Domingo 3 octubre:

- Se realizó la propuesta de diseño del proyecto.

Martes 5 y miércoles 6 octubre:

- Se procedió a agregar las capas necesarias y los paquetes de las capas.
- Para estos días se investigaron problemas de dependencias presentados.

Miércoles 13 octubre:

- Para este día se vieron problemas con Tensorflow lite, por lo que se investigó y se llegó a posibles soluciones mediante la página:

Tsukamoto, N. (2020). *meta-tensorflow-lite: Yocto layer for TensorFlow Lite interpreter with Python*. Recuperado de: <https://github.com/NobuoTsukamoto/meta-tensorflow-lite>

- Además, se investigó sobre otras posibles aplicaciones que no hicieran uso de *Keras* ya que *yocto* no lo soporta.

Viernes 15 y sábado 16 octubre:

- Para este día se vieron errores presentados con tensorflow lite debido a que a la hora de hacer *bitbake python3-tensorflow-lite* se produce el siguiente error:

```
ERROR: ParseError at /home/susana/Descargas/poky-hardknott-25.0.2/meta-openembedded/meta-oe/recipes-extended/socketcan/can-ut
ils_git.bb:51: unparsed line: 'RRECOMMENDS:${PN} += "lproute2"'
```

Este error se logró solucionar borrando y volviendo a descargar mediante un git clone la carpeta poky de yocto project.

Jueves 14 octubre:

- Se investigaron otras posibles aplicaciones y se escogió la siguiente aplicación como la aplicación a trabajar:

Fahrudin, H. (2018). *FastEmotRecognition*. Recuperado de: <https://github.com/hfahrudin/FastEmotRecognition>

Lunes 18 octubre:

- Se realiza prueba de cámara para poder probar la aplicación.
- Para este día se probó la aplicación para verificar su funcionalidad.
- Se realizan las modificaciones de tensorflow a tensorflow lite en la aplicación, con ayuda de la página:

TensorFlow. (2021). Python quickstart. Recuperado de: <https://www.tensorflow.org/lite/guide/python>

- Se crea la receta para la aplicación.

Sábado 23 octubre:

- Se crean dos scripts para poder instalar Yocto Project en la imagen de la raspberry pi, esto para poder descargar dentro de la raspberry pi todos los archivos y correr la aplicación.
- Se añaden estos scripts a la receta de la aplicación.
- Una vez esto se crea una imagen.
- Luego se procede a instalar la imagen creada en la raspberry pi.
- Se prueban los scripts en la raspberry pi.

Domingo 24 octubre:

- Se procede a probar la aplicación en la raspberry pi, por lo que al correr la aplicación en la raspberry pi se presentaron los siguientes errores:

ERROR: Could not find a version that satisfies the requirement scikit-image (from versions: none)

ERROR: No matching distribution found for sckit-image

Estos errores se debían a la fecha que tenía la raspberry, por lo que para solucionar estos errores se ajustó correctamente la fecha a la raspberry.

- Se probó y configuró la cámara de la raspberry pi, en este pasose presentaron los siguientes warnings:

```
[WARN:0] global /usr/src/debug/opencv/4.5.2-r0/git/modules/videoio/src/cap_gstreamer.cpp (597) isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created
```

Este error se arregló agregando la capa gstreamer y agregando cv2.CAP_V4L2 a la línea de comando VideoCapture de código de la aplicación por lo que se ve de la siguiente forma cap = cv2.VideoCapture(0, cv2.CAP_V4L2)

También se presentó el siguiente warning:

```
[WARN:0] global /usr/src/debug/opencv/4.5.2-r0/git/modules/videoio/src/cap_v4l.cpp (890) open VIDEOIO(V4L2: /dev/video10): can't open camera by index
```

Para este error se intentaron varias soluciones pero no se logró solucionar, se intentó solucionar con cv2.VideoCapture(-1, cv2.CAP_V4L2), cv2.VideoCapture(cv2.CAP_V4L2), sudo modprobe bcm2835-v4l2

- Se documentó la estructura del repositorio en el readme.md.
- Se realizó el video demostrativo y bitácoras.