

# Vectores característicos de la librería `face_recognition`

La función encargada de convertir una imagen en una serie de vectores característicos es la siguiente:

```
def face_encodings(face_image, known_face_locations=None, num_jitters=1, model="small"):
    """
    Given an image, return the 128-dimension face encoding for each face in the image.

    :param face_image: The image that contains one or more faces
    :param known_face_locations: Optional - the bounding boxes of each face if you already know them.
    :param num_jitters: How many times to re-sample the face when calculating encoding. Higher is more accurate, but slower (i.e. 100
    :param model: Optional - which model to use. "large" or "small" (default) which only returns 5 points but is faster.
    :return: A list of 128-dimensional face encodings (one for each face in the image)
    """
    raw_landmarks = _raw_face_landmarks(face_image, known_face_locations, model)
    return [np.array(face_encoder.compute_face_descriptor(face_image, raw_landmark_set, num_jitters)) for raw_landmark_set in raw_landmarks]
```

La función `_raw_face_landmarks` hace lo siguiente:

```
def _raw_face_landmarks(face_image, face_locations=None, model="large"):
    if face_locations is None:
        face_locations = _raw_face_locations(face_image)
    else:
        face_locations = [_css_to_rect(face_location) for face_location in face_locations]

    pose_predictor = pose_predictor_68_point

    if model == "small":
        pose_predictor = pose_predictor_5_point

    return [pose_predictor(face_image, face_location) for face_location in face_locations]
```

Se hace uso de la función `pose_predictor` que viene de las siguientes dos instrucciones:

```
predictor_68_point_model = face_recognition_models.pose_predictor_model_location()
pose_predictor_68_point = dlib.shape_predictor(predictor_68_point_model)

predictor_5_point_model = face_recognition_models.pose_predictor_five_point_model_location()
pose_predictor_5_point = dlib.shape_predictor(predictor_5_point_model)
```

La función `dlib.shape_predictor` toma como un input una imagen con una cara humana e identifica las ubicaciones de puntos faciales importantes, como las comisuras de la boca y los ojos, la punta de la nariz, etc.

En conclusión `_raw_face_locations` retorna una lista con la ubicaciones importantes de la cara de una persona.

Por cada una de estas caras se convierte sus ubicaciones importantes mediante un descriptor para crear un vector característico de 128 dimensiones.