def map\_filename\_to\_image\_and\_mask(t\_filename, a\_filename, height=224, width=224):

  '''

  Preprocesses the dataset by:

    \* resizing the input image and label maps

    \* normalizing the input image pixels

    \* reshaping the label maps from (height, width, 1) to (height, width, 12)

  Args:

    t\_filename (string) -- path to the raw input image

    a\_filename (string) -- path to the raw annotation (label map) file

    height (int) -- height in pixels to resize to

    width (int) -- width in pixels to resize to

  Returns:

    image (tensor) -- preprocessed image

    annotation (tensor) -- preprocessed annotation

  '''

  # Convert image and mask files to tensors

  img\_raw = tf.io.read\_file(t\_filename)

  anno\_raw = tf.io.read\_file(a\_filename)

  image = tf.image.decode\_jpeg(img\_raw)

  annotation = tf.image.decode\_jpeg(anno\_raw)

  # Resize image and segmentation mask

  image = tf.image.resize(image, (height, width,))

  annotation = tf.image.resize(annotation, (height, width,))

  image = tf.reshape(image, (height, width, 3,))

  annotation = tf.cast(annotation, dtype=tf.int32)

  annotation = tf.reshape(annotation, (height, width, 1,))

  stack\_list = []

  # Reshape segmentation masks

  for c in range(len(class\_names)):

    mask = tf.equal(annotation[:,:,0], tf.constant(c))

    stack\_list.append(tf.cast(mask, dtype=tf.int32))

  annotation = tf.stack(stack\_list, axis=2)

  # Normalize pixels in the input image

  image = image/127.5

  image -= 1

  return image, annotation