See playlist (https://www.youtube.com/playlist?list=PLQY2H8rRoyvxtOJfAzfs0Cq5FqBuYO8e8)

tf.keras.applications.vgg19.preprocess_input

TensorFlow 1 version (/versions/r1.15/api_docs/python/tf/keras/applications/vgg19/preprocess_input)

View source on GitHub (https://github.com/tensorflow/tensorflow/blob/r1.15/tensorflow/python/keras/applications/vgg19/__init__.py#L234)

Preprocesses a tensor or Numpy array encoding a batch of images.

View aliases

Compat aliases for migration

See Migration guide (https://www.tensorflow.org/guide/migrate) for more details.

`tf.compat.v1.keras.applications.vgg19.preprocess_input` (https://www.tensorflow.org/api_docs/python/tf/keras/applications/vgg19/preprocess_input)

```
tf.keras.applications.vgg19.preprocess_input(
    x, data_format=None
)
```

Used in the notebooks

Used in the tutorials

- Neural style transfer (https://www.tensorflow.org/tutorials/generative/style_transfer)

Usage example with `applications.MobileNet` (https://www.tensorflow.org/api_docs/python/tf/keras/applications/MobileNet):

```
i = tf.keras.layers.Input([None, None, 3], dtype = tf.uint8)
x = tf.cast(i, tf.float32)
x = tf.keras.applications.mobilenet.preprocess_input(x)
core = tf.keras.applications.MobileNet()
x = core(x)
model = tf.keras.Model(inputs=[i], outputs=[x])

image = tf.image.decode_png(tf.io.read_file('file.png'))
result = model(image)
```

Args

x	A floating point `numpy.array` or a `tf.Tensor` (https://www.tensorflow.org/api_docs/python/tf/Tensor), 3D or 4D with 3 color channels, with values in the range [0, 255]. The preprocessed data are written over the input data if the data types are compatible. To avoid this behaviour, `numpy.copy(x)` can be used.
`data_format`	Optional data format of the image tensor/array. Defaults to None, in which case the global setting `tf.keras.backend.image_data_format()` (https://www.tensorflow.org/api_docs/python/tf/keras/backend/image_data_format) is used (unless you changed it, it defaults to "channels_last").

Returns

Preprocessed `numpy.array` or a `tf.Tensor` (https://www.tensorflow.org/api_docs/python/tf/Tensor) with type `float32`.

The images are converted from RGB to BGR, then each color channel is zero-centered with respect to the ImageNet dataset, without scaling.

Raises

`ValueError`	In case of unknown `data_format` argument.