Nama : Andhika Aria Pratama N

NIM : 1103202121

08 PyTorch Paper Replicating

- Menyiapkan pengaturan

Mengimport beberapa library yang akan dipakai

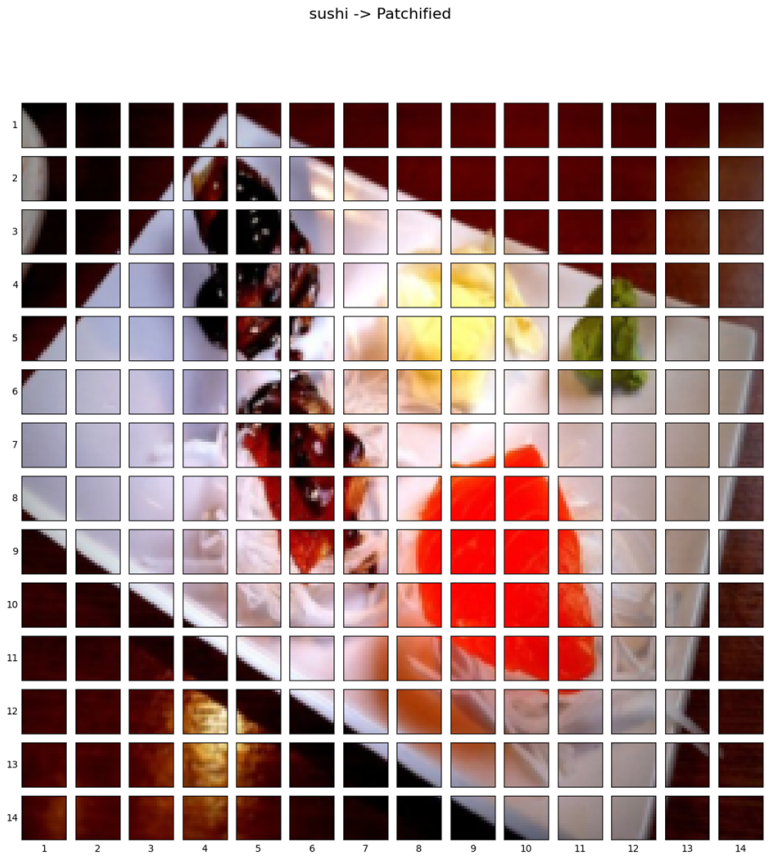
- Mengunduh data

Mengunduh data dari github dan mengatur file train dan test untuk menyimpan data yang telah diunduh.

- Persamaan 1: Membagi data menjadi beberapa patch dan membuat kelas, posisi, dan penyematan patch

Pada tahap ini gambar dihitung jumlah patch (sub-bagian) dalam satu gambar dengan ukuran tinggi (H=224) dan lebar (W=224) serta ukuran patch (P=16). Hasilnya adalah 196 patch. Selanjutnya, ditampilkan bentuk input (shape) untuk lapisan embedding, yang merupakan representasi gambar dengan setiap patch di-flatten menjadi vektor. Terakhir, sebuah contoh gambar dari dataset ditampilkan bersama dengan label kelasnya.

```
# Setup hyperparameters and make sure img_size and patch_size are compatible
img_size = 224
patch_size = 16
num_patches = img_size/patch_size
assert img_size % patch_size == 0, "Image size must be divisible by patch size"
print(f"Number of patches per row: {num_patches}\
        \nNumber of patches per column: {num_patches}\
        \nTotal patches: {num_patches*num_patches}\
        \nPatch size: {patch_size} pixels x {patch_size} pixels")

# Create a series of subplots
fig, axs = plt.subplots(nrows=img_size // patch_size, # need int not float
                        ncols=img_size // patch_size,
                        figsize=(num_patches, num_patches),
                        sharex=True,
                        sharey=True)

# Loop through height and width of image
for i, patch_height in enumerate(range(0, img_size, patch_size)): # iterate through height
    for j, patch_width in enumerate(range(0, img_size, patch_size)): # iterate through width

        # Plot the permuted image patch (image_permuted -> (Height, Width, Color Channels))
        axs[i, j].imshow(image_permuted[patch_height:patch_height+patch_size, # iterate through height
                                        patch_width:patch_width+patch_size, # iterate through width
                                        :]) # get all color channels

        # Set up label information, remove the ticks for clarity and set labels to outside
        axs[i, j].set_ylabel(i+1,
                             rotation="horizontal",
                             horizontalalignment="right",
                             verticalalignment="center")
        axs[i, j].set_xlabel(j+1)
        axs[i, j].set_xticks([])
        axs[i, j].set_yticks([])
        axs[i, j].label_outer()

# Set a super title
fig.suptitle(f"{class_names[label]} -> Patchified", fontsize=16)
plt.show()
```



Kode ini mempersiapkan dan mengecek kecocokan parameter hyperparameter seperti ukuran gambar (img_size), ukuran patch (patch_size), dan jumlah patch dalam satu baris atau kolom (num_patches). Selanjutnya, kode ini membuat dan menampilkan serangkaian subplot yang merepresentasikan gambar input yang tersegmentasi menjadi patch sesuai dengan parameter yang telah diatur. Hasil outputnya seperti pada gambar diatas, gambar diatas adalah tampilan subplot yang menunjukkan gambar input yang tersegmentasi menjadi patch dengan label kelasnya.