Exercise 2: Image Classification with Vision Transformer

Patch Encoder

```
def projection(projection_dim, input_shape):
    return layers.Dense(projection_dim, input_shape=input_shape)

def position_embedding(num_positions, output_dim):
    return layers.Embedding(input_dim=num_positions, output_dim=output_dim)

class PatchEncoder(layers.Layer):
    def __init__(self, projection_dim, **kwargs):
        super(PatchEncoder, self).__init__(**kwargs)
        self.projection_dim = projection_dim

    def build(self, input_shape):
        self.num_patches = input_shape[1]
        raw_patch_dim = input_shape[2] * input_shape[3] * input_shape[4]
        self.reshape = layers.Reshape((self.num_patches, raw_patch_dim))
        self.projection = projection(self.projection_dim, input_shape=(self.num_patches, raw_patch_dim))
        self.position_embedding = position_embedding(num_positions=self.num_patches, output_dim=self.projection_dim)

    def call(self, patch):
        patch = self.reshape(patch)
        patch = self.projection(patch)
        positions = tf.range(start=0, limit=self.num_patches, delta=1)
        encoded = patch + self.position_embedding(positions)
        return encoded
```