1. **A**. The main point for difference between MLP models (fully connected) and CNN models in their numbers of parameters, is for their localities. CNN models which are developed by convolution technique, just create dependency between near pixels in each of their layers. In the MLP model, we need to create connections between all neurons in each layer, we will have much more weights and parameters in this type of model.
We can demonstrate that, based on below reasons, CNN models will have better performance, compared to the MLP models when we work on image processing areas:

1.MLPs have much more parameters compared to CNN and based on that, their operating load will be higher that CNN model; As a consequence of that, in the same time frame, we can make more progress in CNN models.

2. MLP models ignore all spatial features which are one of the important parts that developed in CNN model; Based on this omitting, MLPs will have lower accuracy than CNN.

B. By applying padding with size of 2 on each side, our output will be (16*16) * 16 and we will have 2016 parameters in these layers (we assumed that our convolution matrix is (5*5) * 5).

C.

Input shape	Conv2D	Output shape
(32*32) * 3	(5*5) * 3	(28*28) * 3
(32*32) * 3	(3*3) * 9	(30*30) * 9
(32*32) * 3	(3*3) * 9	(28*28) * 9

D. Let's start our scenario by comparing “max pooling” and “Min pooling”.
These are the two polling which we used in most of our works. In most cases, we use max pooling in images which have dark backgrounds like MNIST (digits are white and backgrounds are black) and we use min pooling when our background is brighter than our objects. Average pooling is a mix of both of them; In average pooling we try to smooth our image.
Global average pooling used for replacing FC and dense layers in our models. The model ends with a convolutional layer that generates as many feature maps as the number of target classes, and applies global average pooling to each in order to convert each feature map into one value.
Based on **Rahman** (n.d.) idea, we don’t think average pooling has any significant advantage over max-pooling. But, may be in some cases, where variance in a max pool filter is not significant, both pooling will give same type results. But in extreme cases, max-pooling will provide better results for sure.
I would add an additional argument - that max-pooling layers are worse at preserving localization.
E. The VGG has a small filter with much more layers compared to older models. In VGG, all of the filters were 3*3(with stride = 1) and it has 16 or 19 layers.
The ResNet has a lot of “reminder” blocks which each of these blocks has 2 convolution layers with 3*3 filters. Generally, the number of filters is doubled and the spatial resolution is halved. First, it has a convolutional layer. After the last remaining block, dimension the data using Pooling Average is reduced and an FC layer is used for classification.
For the ImageNet problem, different network depths include 34, 50, 101 and 152
have been used.
In this model, we have 1*1 filters which are used for decreasing our depth
We have 2 facts which cause to have better speed in ResNet compared to VGG models:

- In first 4 layers of VGG, we have 2 conv2D with size of (3*3) *64 and 2 conv2D with size of (3*3) *128 which after these 4 layers our output shape will size of 56. All of this action was done in ResNet model with just one conv2D with size of (7*7) *64, /2 which is the most important part that cause to have better speed in Resnet.
- After these 4 layers, In VGG, we have 4 Conv2D with size of (3*3) * 256 and in same position, in ResNet, we have 6 conv2D with size of (3*3) * 64. We have more layers in ResNet but their filters are smaller and based on that, we have better speed in the part too.
- Also, we use (1*1) filters in ResNet for decreasing our depth which is one of the reasons for this question.

The idea of ResNet is that instead of learning the desired mapping, the layers of the network learn its remainder and in the VGG, the idea that we can use more layers with smaller convolution size.

2. **A**. In most cases, by decreasing loss, we can see that our accuracy will grow up; However, we don't have any mathematical relations between these two metrics.

Model type	Accuracy	Loss
CNN	65.75 %	1.0829
MLP	41.60 %	1.6250

B. On average, in a dense layer, the execution period was 6.8 second and for CNN model, it was 8.2 seconds.

C. Number of parameters reduces the amount of space required to store the network, but it doesn't mean that it's faster.

3. By using pre-trained model with ImageNet weights, we reached accuracy equal to 28.46% and by just removing ImageNet weights and train model with initial weights we could reach accuracy equal to 9.45%. Based on this, we can demonstrate that, by using pre-trained models with their weights, we can reach higher accuracy in the same amount of time in our training procedure.

Reference:

machine-learning-articles/what-are-max-pooling-average-pooling-global-max-pooling-and-global-average-pooling.md at main · christianversloot/machine-learning-articles · GitHub

Maxpooling vs minpooling vs average pooling | by Madhushree Basavarajaiah | Medium