**VGG-16 PAPER POINTS:**

* Thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution ﬁlters, which shows that a signiﬁcant improvement on the prior-art conﬁgurations can be achieved by pushing the depth to 16–19 weight layers.
* We ﬁx other parameters of the architecture, and steadily increase the depth of the network by adding more convolutional layers, which is feasible due to the use of very small (3 × 3) convolution ﬁlters in all layers.
* During training, the input to our ConvNets is a ﬁxed-size 224 × 224 RGB image. The only preprocessing we do is subtracting the mean RGB value, computed on the training set, from each pixel.
* we use ﬁlters with a very small receptive ﬁeld: 3 × 3 (which is the smallest size to capture the notion of left/right, up/down,centre)
* The convolution stride is ﬁxed to 1 pixel,. the padding is 1 pixel for 3 × 3 conv. layers, Max-pooling is performed over a 2 × 2 pixel window, with stride 2
* A stack of convolutional layers (which has a different depth in different architectures) is followed by three Fully-Connected (FC) layers: the ﬁrst two have 4096 channels each, the third which contains 1000 channels (one for each class). The ﬁnal layer is the soft-max layer. The conﬁguration of the fully connected layers is the same in all networks.
* All hidden layers are equipped with the rectiﬁcation ,none of our networks (except for one) contain Local Response Normalisation (LRN)
* Rather than using relatively large receptive ﬁelds in the ﬁrst conv. layers, or 7×7 with stride 2 in,we use very small 3 × 3 receptive ﬁelds throughout the whole net.
* The incorporation of 1 × 1 conv. layers (conﬁguration C, Table 1) is a way to increase the nonlinearity of the decision function without affecting the receptive ﬁelds of the conv layers.
* The batch size was set to 256, momentum to 0.9. The training was regularised by weight decay (the L2 penalty multiplier set to 5·10^−4) and dropout regularisation for the ﬁrst two fully-connected layers (dropout ratio set to 0.5). The learning rate was initially set to 10^−2, and then decreased by a factor of 10 when the validation set accuracy stopped improving.
* The initialisation of the network weights is important, since bad initialisation can stall learning due to the instability of gradient in deep nets. To circumvent this problem, we began with training the conﬁguration A (Table 1), shallow enough to be trained with random initialisation. Then, when training deeper architectures, we initialised the ﬁrst four convolutional layers and the last three fully connected layers with the layers of net A (the intermediate layers were initialised randomly). We did not decrease the learning rate for the pre-initialised layers, allowing them to change during learning
* For random initialisation (where applicable), we sampled the weights from a normal distribution with the zero mean and 10−2 variance. The biases were initialised with zero
* At test time,the fully-connected layers are ﬁrst converted to convolutional layers (the ﬁrst FC layer to a 7 × 7 conv. layer, the last two FC layers to 1 × 1 conv. layers). The resulting fully-convolutional net is then applied to the whole (uncropped) image. The result is a class score map with the number of channels equal to the number of classes, and a variable spatial resolution, dependent on the input image size. Finally, to obtain a ﬁxed-size vector of class scores for the image, the class score map is spatially averaged (sum-pooled). We also augment the test set by horizontal ﬂipping of the images; the soft-max class posteriors of the original and ﬂipped images are averaged to obtain the ﬁnal scores for the image.
* Since the fully-convolutional network is applied over the whole image, there is no need to sample multiple crops at test time (Krizhevsky et al., 2012), which is less efﬁcient as it requires network re-computation for each crop
* when applying a ConvNet to a crop, the convolved feature maps are padded with zeros, while in the case of dense evaluation the padding for the same crop naturally comes from the neighbouring parts of an image (due to both the convolutions and spatial pooling), which substantially increases the overall network receptive ﬁeld, so more context is captured
* . Multi-GPU training exploits data parallelism, and is carried out by splitting each batch of training images into several GPU batches, processed in parallel on each GPU. After the GPU batch gradients are computed, they are averaged to obtain the gradient of the full batch. Gradient computation is synchronous across the GPUs, so the result is exactly the same as when training on a single GPU.
* we have found that our scheme provides a speedup of 3.75 times on an off-the-shelf 4-GPU system, as compared to using a single GPU
* SINGLE SCALE EVALUATION:
  + First, we note that using local response normalisation (A-LRN network) does not improve on the model A without any normalisation layers. We thus do not employ normalisation in the deeper architectures (B–E).
  + Second, we observe that the classiﬁcation error decreases with the increased ConvNet depth
  + While the additional non-linearity does help (C has extra 1x1 conv)(C is better than B), it is also important to capture spatial context by using conv. filters with non-trivial receptive ﬁelds (D replaces it with 3x3)(D is better than C).
  + a deep net with small ﬁlters outperforms a shallow net with larger ﬁlters.
  + training set augmentation by scale jittering is indeed helpful for capturing multi-scale image statistics
* MULTI-SCALE EVALUATION
  + the models trained with ﬁxed S were evaluated over three test image sizes, close to the training one: Q = {S − 32,S,S + 32}. At the same time, scale jittering at training time allows the network to be applied to a wider range of scales at test time, so the model trained with variable S ∈ [Smin;Smax] was evaluated over a larger range of sizes Q = {Smin,0.5(Smin + Smax),Smax}.
  + scale jittering at test time leads to better performance (as compared to evaluating the same model at a single scale)
  + we compare dense ConvNet evaluation with multi-crop evaluation,using multiple crops performs slightly better than dense evaluation, and the two approaches are indeed complementary, as their combination outperforms each of them. As noted above, we hypothesise that this is due to a different treatment of convolution boundary conditions.
  + we combine the outputs of several models by averaging their soft-max class posteriors. This improves the performance due to complementarity of the models
* The resulting ensemble of 7 networks has 7.3% ILSVRC test error. After the submission, we considered an ensemble of only two best-performing multi-scale models (conﬁgurations D and E), which reduced the test error to 7.0% using dense evaluation and 6.8% using combined dense and multi-crop evaluation. For reference, our best-performing single model achieves 7.1% error (model E)
* CONCLUSION: It was demonstrated that the representation depth is beneﬁcial for the classiﬁcation accuracy, and that state-of-the-art performance on the ImageNet challenge dataset can be achieved using a conventionalConvNet architecture (LeCun et al., 1989; Krizhevsky et al., 2012) with substantially increased depth

**LOCALISATION**

* It can be seen as a special case of object detection, where a single object bounding box should be predicted for each of the top-5 classes, irrespective of the actual number of objects of the class
* To perform object localisation, we use a very deep ConvNet, where the last fully connected layer predicts the bounding box location instead of the class scores. A bounding box is represented by a 4-D vector storing its center coordinates, width, and height
* There is a choice of whether the bounding box prediction is shared across all classes (single-class regression, SCR) or is class-speciﬁc (per-class regression, PCR). In the former case, the last layer is 4-D, while in the latter it is 4000-D (since there are 1000 classes in the dataset)
* Apart from the last bounding box prediction layer, we use the ConvNet architecture D which contains 16 weight layers and was found to be the best-performing in the classiﬁcation task
* We replace the logistic regression objective with a Euclidean loss, which penalises the deviation of the predicted bounding box parameters from the ground-truth)
* Training was initialised with the corresponding classiﬁcation models (trained on the same scales), and the initial learning rate was set to 10−3. The last fully-connected layer was initialised randomly and trained from scratch.
* Output of the last fully-connected layer is a set of bounding box predictions. To come up with the ﬁnal prediction, we utilise the greedy merging procedure of Sermanet et al. (2014), which ﬁrst merges spatially close predictions (by averaging their coordinates), and then rates them based on the class scores, obtained from the classiﬁcation ConvNet
* ILSVRC criterion : the bounding box prediction is deemed correct if its intersection over union ratio with the ground-truth bounding box is above 0.5.
* ﬁne-tuning all layers for the localisation task leads to noticeably better results than ﬁne-tuning only the fully-connected layers
* application of the localisation ConvNet to the whole image substantially improves the results compared to using a center crop, testing at several scales and combining the predictions of multiple networks further improves the performance
* performance advancement is brought by our very deep ConvNets – we got better results with a simpler localisation method, but a more powerful representation.
* To utilise the ConvNets, pre-trained on ILSVRC, for image classiﬁcation on other datasets, we remove the last fully-connected layer (which performs 1000-way ILSVRC classiﬁcation), and use 4096-D activations of the penultimate layer as image features, which are aggregated across multiple locations and scales. The resulting image descriptor is L2-normalised and combined with a linear SVM classiﬁer, trained on the target dataset. For simplicity, pre-trained ConvNet weights are kept ﬁxed (no ﬁne-tuning is performed).