Supplementary Material

Next, we present the Supplementary Material for the paper "Gabor Layers Enhance Network Robustness". In Section 6 we report the details for the choice of the p parameter of the Gabor layers for the various experiments we present in the paper; Section 7 shows the flip rates for all the experiments we reported; Section 8 presents the results for one experiment on ImageNet [40], both in terms of adversarial accuracy and flip rate; in Section 9 we present several comparisons of distributions of singular values of standard convolutional layers *vs.* Gabor layers; finally, Section 10 depicts visualizations of the filters learned in the Gabor layers when various architectures are trained on numerous datasets.

6 Number of families p search space

We report the details for the search for the p parameter in Tables 5 through 11.

Table 5: **Search Space for LeNet on MNIST** Test set accuracies and flip rates on MNIST with LeNet. Only the first convolutional layer was enhanced with p families.

	Standard		$p = 2$		$p = 3$	
ϵ	Accuracy	Flip Rate	Accuracy	Flip Rate	Accuracy	Flip Rate
0	99.22	-	99.03	-	99.10	-
0.1	80.04	19.53	80.58	18.88	62.98	36.54
0.2	4.39	95.47	7.94	91.85	2.22	97.67
0.3	0.44	99.7	0.78	99.24	0.41	99.68

Table 6: **Search Space for AlexNet on CIFAR100.** Test set accuracies comparison with the standard AlexNet and his enhanced versions. Only the first convolutional layer was enhanced. S stands for standard.

	Accuracy								
ϵ	S	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$	$p = 9$
0	46.48	42.14	42.81	41.99	43.19	42.32	45.15	42.69	43.36
$^2/_{255}$	15.08	11.84	13.01	13.41	13.62	13.55	14.77	12.82	13.16
$^8/_{255}$	4.80	6.82	7.40	6.74	7.81	7.01	7.71	7.86	7.73
$^{16}/_{255}$	5.37	5.92	6.12	5.75	6.23	5.34	6.25	6.07	6.29

Table 7: **Search Space for VGG16 on CIFAR100 – First Layer.** Test set accuracies comparison with the standard VGG16 and his enhanced versions. The first Gabor layer number of families p_1 is explored. S stands for standard.

	Accuracy						
ϵ	S	$p_1 = 1$	$p_1 = 2$	$p_1 = 3$	$p_1 = 4$	$p_1 = 5$	$p_1 = 6$
0	67.54	65.82	67.35	67.05	68.65	68.10	66.23
$2/255$	27.22	28.35	27.89	27.31	27.83	27.32	24.83
$8/255$	18.46	23.05	20.78	20.05	20.66	19.44	18.95
$16/255$	10.49	13.92	12.18	11.30	11.02	10.64	11.29

Table 8: **Search Space for VGG16 on CIFAR100 – Second Layer.** Test set accuracies comparison with the standard VGG16 and his enhanced versions. The first convolutional layer is modified with $p_1 = 3$ families. The second Gabor layer number of families p_2 is explored. S stands for standard.

	Accuracy						
ϵ	S	$p_1 = 3$	$p_2 = 1$	$p_2 = 2$	$p_2 = 3$	$p_2 = 4$	$p_2 = 5$
0	67.54	67.05	64.10	67.68	66.67	65.06	64.81
$2/255$	27.22	27.31	31.36	16.57	27.43	28.25	27.92
$8/255$	18.46	20.05	26.09	10.41	21.15	22.50	22.00
$16/255$	10.49	11.30	14.97	5.10	11.30	12.64	13.11

Table 9: **Search Space for VGG16 on CIFAR100 – Third Layer.** Test set accuracies comparison with the standard VGG16 and his enhanced versions. The first and second convolutional layers are modified with $p_1 = 3$ and $p_2 = 1$ families respectively. The third Gabor layer number of families p_3 is explored. S stands for standard.

	Accuracy			
ϵ	S	$p_1 = 3, p_2 = 1$	$p_3 = 2$	$p_3 = 3$
0	67.54	64.10	63.74	64.49
$2/255$	27.22	31.36	17.13	31.12
$8/255$	18.46	26.09	14.67	25.82
$16/255$	10.49	14.97	9.20	15.40

Table 10: **Search Space for Wide-ResNet on SVHN – First Layer.** Test set accuracies comparison with the standard Wide-ResNet and his enhanced versions. The first Gabor layer number of families p_1 is explored. S stands for standard.

	Accuracy					
ϵ	S	$p_1 = 2$	$p_1 = 3$	$p_1 = 4$	$p_1 = 5$	$p_1 = 6$
0	96.62	96.93	96.72	96.70	96.65	96.73
$2/255$	40.27	45.84	41.37	49.35	43.56	45.66
$8/255$	1.03	0.93	1.09	1.03	1.08	1.03
$16/255$	1.32	1.11	1.32	1.42	1.38	1.28

Table 11: **Search Space for Wide-ResNet on SVHN – Second Layer.** Test set accuracies comparison with the standard Wide-ResNet and his enhanced versions. The first convolutional layer is modified with $p_1 = 4$ families. The second Gabor layer number of families p_2 is explored. S stands for standard.

	Accuracy					
ϵ	S	$p_1 = 4$	$p_2 = 1$	$p_2 = 2$	$p_2 = 3$	$p_2 = 4$
0	96.62	96.70	96.67	96.71	96.60	96.71
$2/255$	40.27	49.35	41.83	44.50	44.11	44.41
$8/255$	1.03	1.03	1.01	0.99	0.98	1.03
$16/255$	1.32	1.42	1.24	1.31	1.28	1.26

Table 12: **Flip rates comparison**. We compare Standard (S), Gabor-layered (G), and *regularized* Gabor-layered (G+r) architectures. For each attack strength (ϵ), the lowest flip rate is in **bold**; second-lowest is underlined.

ϵ		$2/255$			$8/255$			$16/255$		
Dataset	Network	S	G	G+r	S	G	G+r	S	G	G+r
SVHN	WRN	57.22	48.13	**44.19**	98.30	**98.13**	98.17	99.06	99.02	**99.01**
SVHN	VGG16	39.53	34.11	**32.62**	92.82	83.83	**82.33**	97.40	91.56	**90.78**
CIFAR10	VGG16	60.03	56.42	**55.71**	74.51	67.83	**67.46**	86.47	80.84	**80.18**
CIFAR100	AN	56.88	49.73	**49.15**	81.38	**72.82**	72.87	87.77	**83.00**	83.08
CIFAR100	WRN	82.05	76.52	**74.72**	93.82	92.58	**92.25**	96.94	96.59	**96.35**
CIFAR100	VGG16	57.05	**50.94**	51.05	77.94	68.95	**68.45**	90.48	85.75	**85.16**

Table 13: **Flip rates comparison on MNIST**. We compare Standard (S), Gabor-layered (G), and *regularized* Gabor-layered (G+r) architectures on MNIST. For each attack strength (ϵ), the lowest flip rate is in **bold**; second-lowest is underlined.

ϵ		0.1			0.2			0.3		
Dataset	Network	S	G	G+r	S	G	G+r	S	G	G+r
MNIST	LeNet	19.53	18.88	**11.05**	95.47	91.85	**77.27**	99.70	**99.24**	99.64

7 Flip rates

In our work, we assess the robustness of Deep Neural Networks (DNNs) through adversarial accuracies (reported in the main document) and flip rates. Next, we report the flip rates for the main experiments of the paper. Table 12 shows results on SVHN, CIFAR10 and CIFAR100, and Table 13 presents results on MNIST.

8 ImageNet Results

We conduct adversarial attacks for $\epsilon \in \{8/255, 16/255\}$. In Table 14 we report the adversarial accuracies and flip rates for VGG16 [44] trained on ImageNet [40].

9 Singular Values

We report the distribution of singular values of the filters of up to the first three convolutional layers of several Convolutional Neural Network (CNN) architectures (LeNet [24], AlexNet [21], WideResNet [51], and VGG16 [44]) trained in

Table 14: **Adversarial accuracy and flip rate comparison for VGG16 on ImageNet.** We compare Standard (S) and Gabor-layered (G) architectures. For each attack strength (ϵ), the best performance is in **bold**.

	Adv. Accuracy		**Flip Rate**	
ϵ	S	G	S	G
0	**71.20**	68.90	-	-
$^{8}/_{255}$	**2.95**	2.24	95.07	**94.46**
$^{16}/_{255}$	**3.33**	3.15	97.37	**96.68**

various datasets (MNIST [22], CIFAR10, CIFAR100 [20], and ImageNet [40]). The singular values of the layers are computed following [42,4]. The largest singular value of each layer's filter corresponds to the filters' Lipschitz constant [4].

In each histogram plot we show the distribution of singular values of (1) the standard architecture, in blue, and (2) the Gabor-layered architecture, in orange. For some experiments, we also show the distribution of singular values of the Gabor-layered architecture *with* regularization, in green.

For visualization purposes, we set the upper x-limit of each histogram plot to the 95th percentile of the distribution with the largest maximum value.

Next, we list the dataset-architecture pairs, and the Figures in which its distributions are shown:

- **MNIST-LeNet.** Figure 5.
- **CIFAR100-AlexNet.** Figure 6.
- **CIFAR10-VGG16.** Figures 7 - 9.
- **CIFAR100-VGG16.** Figures 10 - 12.
- **ImageNet-VGG16.** Figures 13 - 15.

In most cases, the distribution of singular values of the Gabor-layered version of the networks tends to be around smaller values, usually with high peaks between 0 and 0.5, than that of the standard network.

In terms of the Lipschitz constant of the layers, in most cases we observe that the Gabor-layered versions of the layers have lower Lipschitz constants, and that applying regularization results in even lower Lipschitz constants.

10 Filter visualizations

We report visualizations of the filters of the first convolutional layer of some of the architectures we experimented with. Figures 16 through 23 depict the filters. For the standard convolutional layers, we visualize each filter as an RGB image, where each "channel" of the filter is scaled to be between 0 and 1.

For the Gabor layers, each filter has 1 channel and, hence, we visualize it as is. We show the filters that share the Gabor function G_θ in the same row; while each column corresponds to one of the 8 α-scaled rotations of the filter.

We report the filters for:

– **MNIST-LeNet.** Figures 16 - 18.
– **CIFAR100-AlexNet.** Figures 19 - 21.
– **ImageNet-VGG16.** Figures 22 - 23.

We note that, for LeNet on MNIST, the Gabor-layered version, without regularization, already has filters that strongly resemble a Dirac-delta function and that, as reported in the paper, this network already shows improvements in terms of robustness. Furthermore, when applying regularization, we observe that all filters in the layer become Dirac-delta functions (see Figure 18). Again, as reported in the paper, regularization showed large gains in robustness.

The filters from AlexNet are useful for visualizing the modeling capabilities of Gabor functions, as shown in Figure 20, where we observe blob-like patterns, and also oriented and scaled edges and bars. These patterns, while simpler than those of the standard convolutional layers (see Figure 19), provide on-pair accuracy with such layers, while also providing gains in robustness. For the case of AlexNet, however, we observe that regularization has virtually no impact in the form of the filters that are learnt (see Figure 21).

In Figure 22 we report the filters learnt by the standard version of VGG16 on ImageNet. Patterns are, however, not straightforward to visualize in these filters. By construction, the filters learnt by the Gabor-layered version of VGG16 when fine-tuned on ImageNet are more visually-appealing, as shown in Figure 23. Note, also, that some Gabor filters have strong similarities between one another.

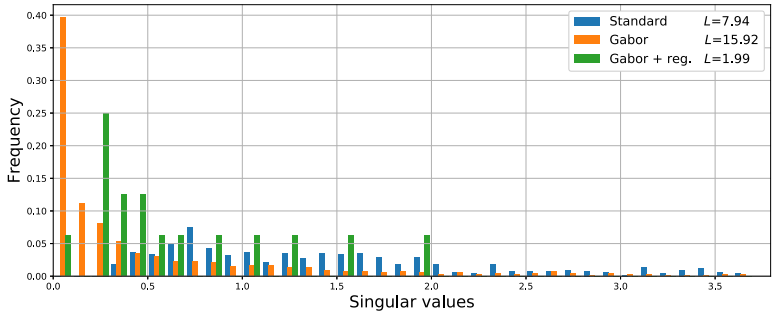

Fig. 5: **Distribution of singular values for the first layer of LeNet trained on MNIST.** The legend shows the largest singular value, *i.e.* the Lipschitz constant of the layer.

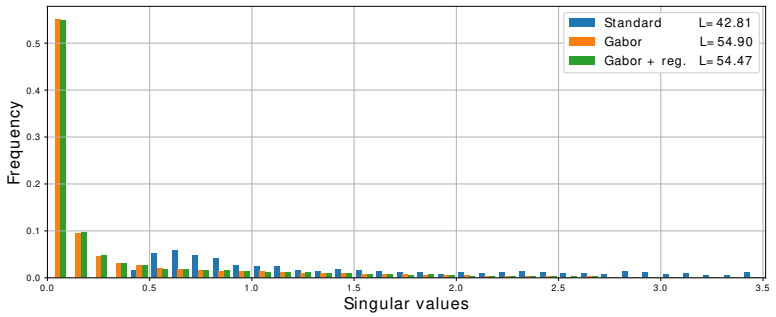

Fig. 6: **Distribution of singular values for the first layer of AlexNet trained on CIFAR100.** The legend shows the largest singular value, *i.e.* the Lipschitz constant of the layer.

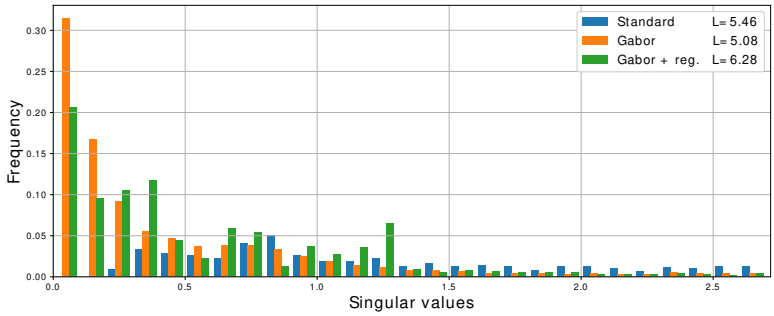

Fig. 7: **Distribution of singular values for the first layer of VGG16 trained on CIFAR10.** The legend shows the largest singular value, *i.e.* the Lipschitz constant of the layer.

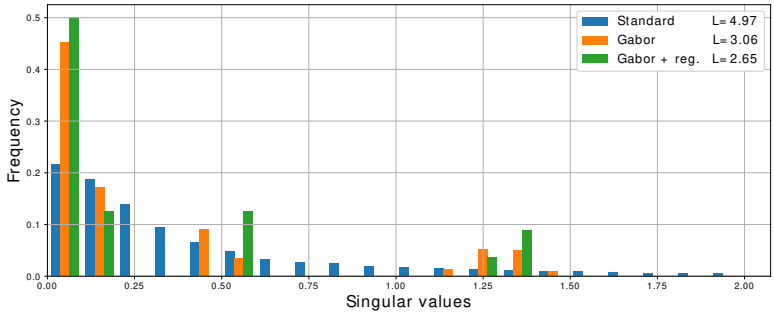

Fig. 8: **Distribution of singular values for the second layer of VGG16 trained on CIFAR10.** The legend shows the largest singular value, *i.e.* the Lipschitz constant of the layer.

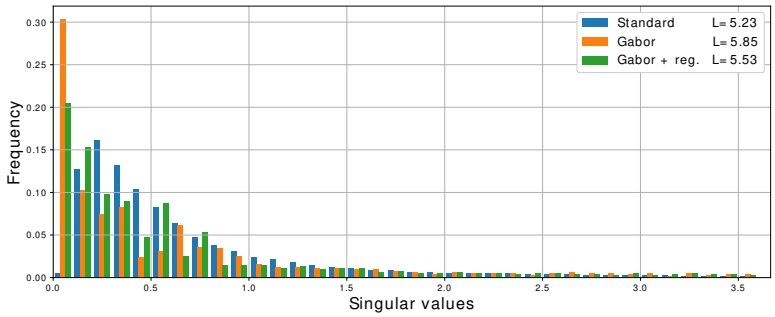

Fig. 9: **Distribution of singular values for the third layer of VGG16 trained on CIFAR10.** The legend shows the largest singular value, *i.e.* the Lipschitz constant of the layer.

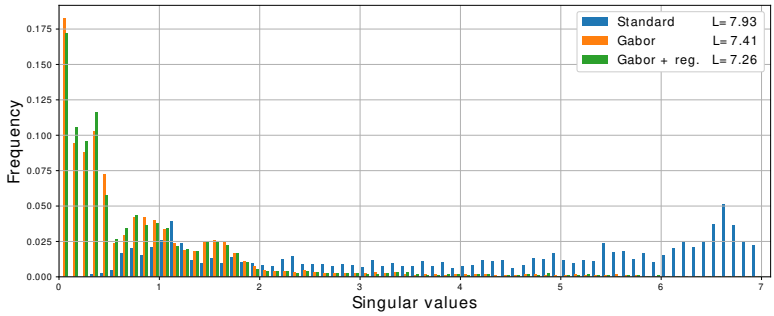

Fig. 10: **Distribution of singular values for the first layer of VGG16 trained on CIFAR100.** The legend shows the largest singular value, *i.e.* the Lipschitz constant of the layer.

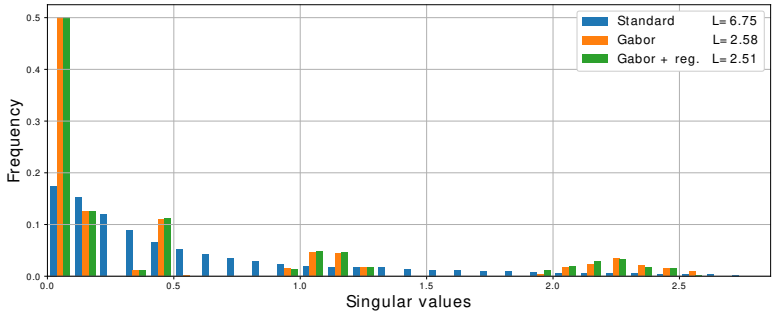

Fig. 11: **Distribution of singular values for the second layer of VGG16 trained on CIFAR100.** The legend shows the largest singular value, *i.e.* the Lipschitz constant of the layer.

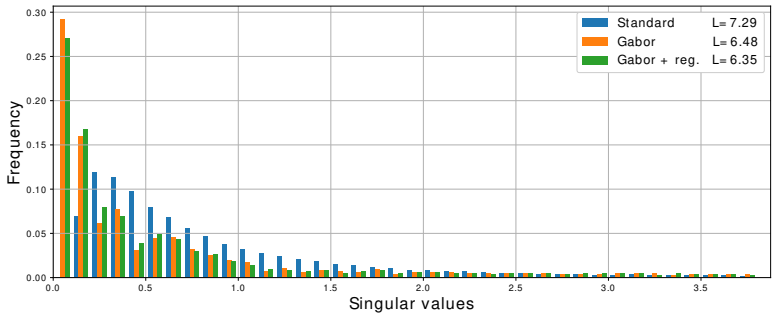

Fig. 12: **Distribution of singular values for the third layer of VGG16 trained on CIFAR100.** The legend shows the largest singular value, *i.e.* the Lipschitz constant of the layer.

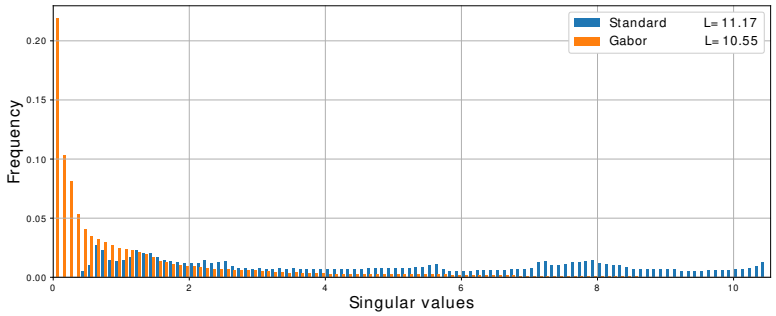

Fig. 13: **Distribution of singular values for the first layer of VGG16 trained on ImageNet.** The legend shows the largest singular value, *i.e.* the Lipschitz constant of the layer. Note that the Gabor-layered version was fine-tuned, starting from ImageNet-pretrained weights, due to computational constraints.

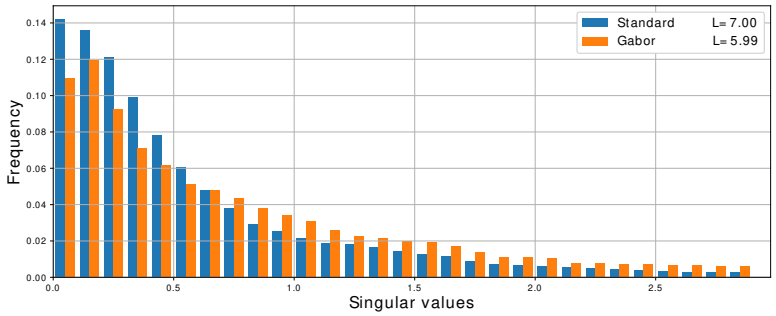

Fig. 14: **Distribution of singular values for the second layer of VGG16 trained on ImageNet.** The legend shows the largest singular value, *i.e.* the Lipschitz constant of the layer. Note that the Gabor-layered version was fine-tuned, starting from ImageNet-pretrained weights, due to computational constraints.

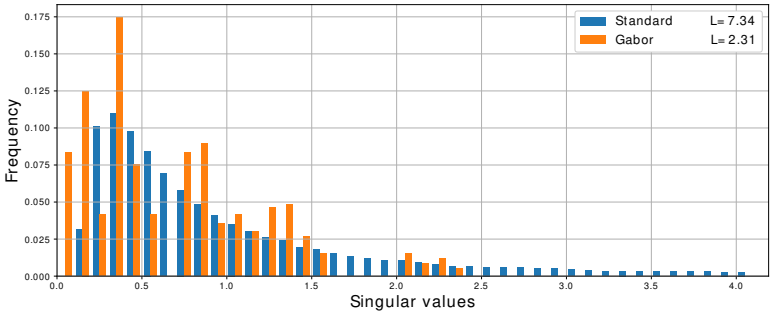

Fig. 15: **Distribution of singular values for the third layer of VGG16 trained on ImageNet.** The legend shows the largest singular value, *i.e.* the Lipschitz constant of the layer. Note that the Gabor-layered version was fine-tuned, starting from ImageNet-pretrained weights, due to computational constraints.

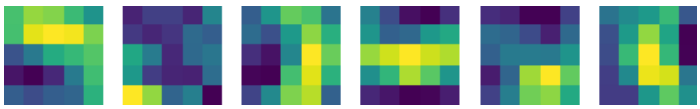

Fig. 16: **Standard LeNet-MNIST filters from the first convolutional layer.** The 6 grayscale filters are visualized.

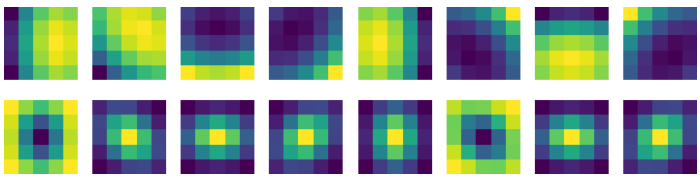

Fig. 17: **Gabor-layered LeNet-MNIST filters from the first convolutional layer.** Each of the rows in the figure corresponds to each of the 7 families of filters of the layer. Each column is one of the 8 α-scaled rotations of the filter. Note that the filters in the second row resemble a Dirac-delta function.

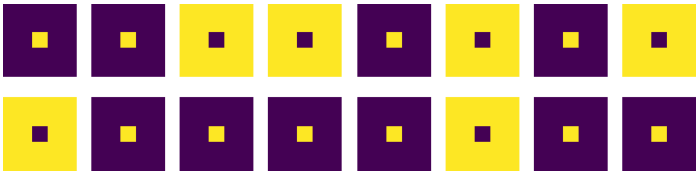

Fig. 18: **Gabor-layered LeNet-MNIST filters from the first convolutional layer with regularization.** Each of the rows in the figure corresponds to each of the 7 families of filters of the layer. Each column is one of the 8 α-scaled rotations of the filter. Note that regularization enforces the filters to be Dirac-deltas and, as reported in the paper, this change results in large gains in robustness.

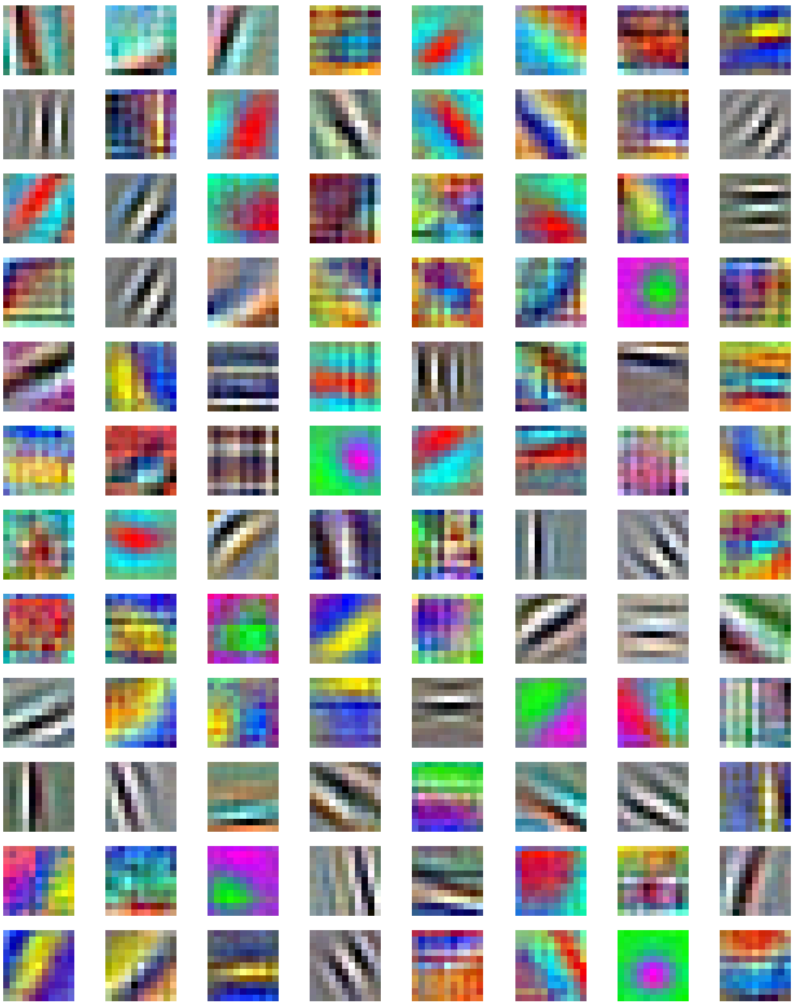

Fig. 19: **Standard AlexNet-CIFAR100 filters from the first convolutional layer.** The 96 filters are visualized by concatenating the RGB channels and mapping values from 0 to 1.

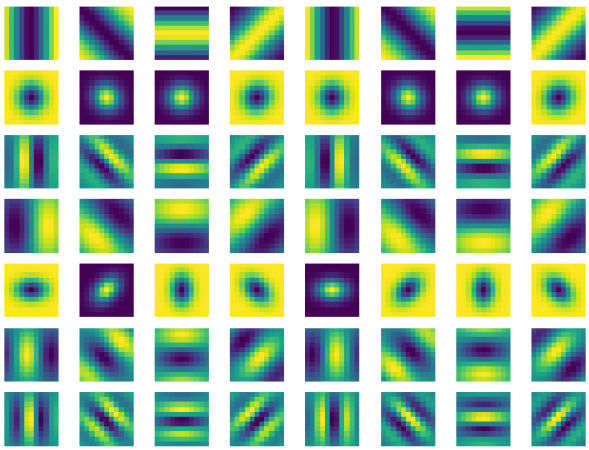

Fig. 20: **Gabor-layered AlexNet-CIFAR100 filters from the first convolutional layer.** Each of the rows in the figure corresponds to each of the 7 families of filters of the layer. Each column is one of the 8 α-scaled rotations of the filter.

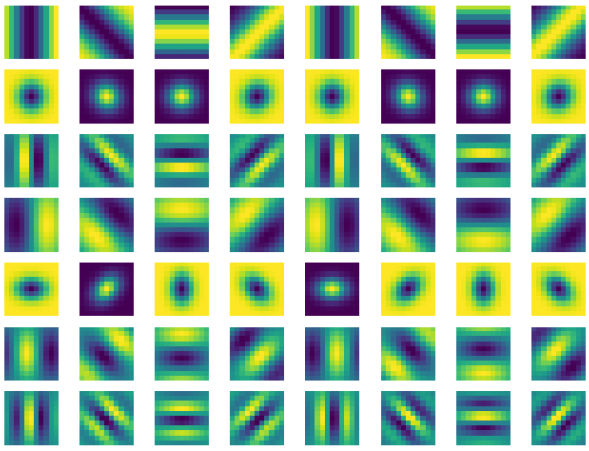

Fig. 21: **Gabor-layered AlexNet-CIFAR100 filters from the first convolutional layer with regularization.** Each of the rows in the figure corresponds to each of the 7 families of filters of the layer. Each column is one of the 8 α-scaled rotations of the filter.

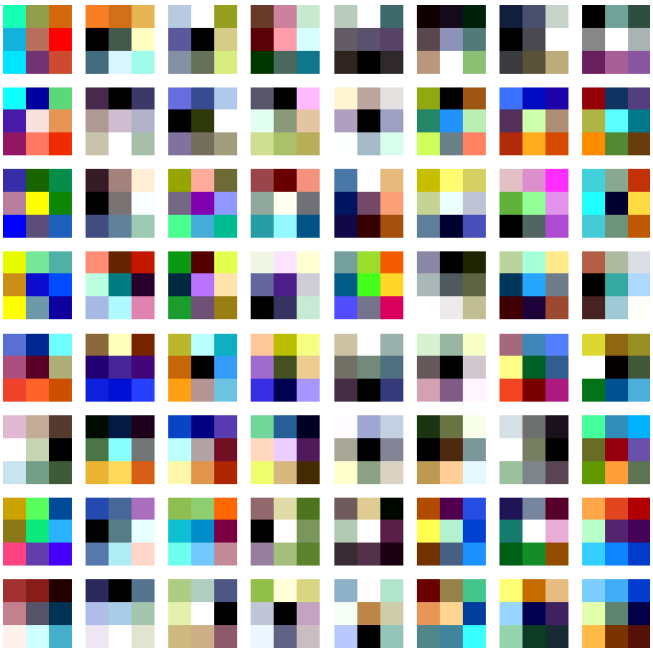

Fig. 22: **Standard VGG16-ImageNet filters from the first convolutional layer.** The 64 filters are visualized by concatenating the RGB channels and mapping values from 0 to 1.

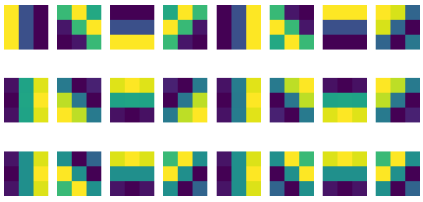

Fig. 23: **Gabor-layered VGG16-ImageNet filters from the first convolutional layer without regularization.** Each of the rows in the figure corresponds to each of the 3 families of filters of the layer. Each column is one of the 8 α-scaled rotations of the filter.