How does CNN grasp transparent object features?

Roland Sireyjol
Kyushu University, Japan
roland@limu.ait.kyushu-u.ac.jp

Atsushi Shimada
Kyushu University, Japan
atsushi@limu.ait.kyushu-u.ac.jp

Tsubasa Minematsu
Kyushu University, Japan
minematsu@limu.ait.kyushu-u.ac.jp

Hajime Nagahara
Osaka University, Japan
nagahara@ids.osaka-u.ac.jp

Rin-ichiro Taniguchi
Kyushu University, Japan
rin@kyudai.jp

***Abstract*—Much attention has been paid to object classification by CNN (convolutional neural network). The CNN has a great ability to grasp efficient features from a large scale of training samples automatically. In this paper, we tackle an issue of transparent object classification, and investigate how the CNN grasp transparent object features.**

***Index Terms*—classification, light field, neural network, distortion**

I. Introduction

Image-capturing devices are becoming very common, along with object classification softwares, to such an extend even a simple cellphone can realise state-of-the-art image processing methods. However, transparent objects do not offer features easy to identify: instead of hiding the background like opaque objects, they merely distort it . Therefore, their appearance drastically change regarding to their environment, and traditionnal methods like SIFT or SURF [1] fail at identifying such objects. This difficulty has been highlighted in a previous paper [2] which proposed to use the light field camera (LFC)'s potential to create the light field distortion (LFD) feature, which uses the distortion made on the background by the object according to its refraction caracteristics [3]. Even though obtained results with this technique were way better than those obtained with other identification methods, with 18 category of objects and optimal conditions, recognition reached 85% accuracy. Moreover, this feature highly depends on various parameters (camera orientation, illumination, number of cluster, etc), and setting those by hand can be very complex. Various methods have tackled the problem of transparent object recognition, through complex capture or processing techiques [4], [5] .

On another hand, convolutional neural networks (CNN) has shown excellent results regarding image processing and object classification. Therefore they seem a viable option to identify transparent objects [6], and might obtain better results than hand made distortion feature. By learning features autonomously, CNN can become a powerful and efficient identification tool, however those features are hard to identify or compare with complex hand made features. In this paper, we study what kind of features such CNN learns at each layer, and establish a methodology to identify if one of those features is similar to LFD, when no preprocessing has been made on the input data.

II. Analytics methodology

A. Light field dataset

The data used for this study was obtained with a ProFUSION-25C [7], capturing 5*5 VGA images simultaneously from different viewpoints. Each image is originally 640*480 pixels, but they have been cropped to 480*432 pixels, and set in black and white.

Data obtained from a LFC extends on four dimensions (s,t,u,v): the viewpoint plane (s,t) can be associated to the position of the camera among the 5*5 ProFusion25 cameras, and the image plane (u,v) can be associated to the usual width and height coordinates of a pixel for each image captured by the ProFusion-25C cameras(cf Fig. 1). 20 different objects were captured in front of 10 backgrounds (One can argue that our dataset is very limited, but the purpose of this study is to establish a methodology to identify what kind of feature is learned).

B. Light field data adaptation to CNN

Despite the fact that LF data extends on four dimensions, a 3D Convolutional Neural Network (CNN) has been used to ease the research process. To adapt the data to such a CNN, frames (u,v) from a single axis (s) has been used, representing five consecutive frames as shown in Fig. 1. Limitating the data this way allows us to process it in a somewhat similar way than a video (time, width, height), and each input data is therefore set as (s,u,v), also described as (depth,width,height). The architecture of our CNN is presented in Fig. 2, (inspired by [8]). We can notice that the original input' scale is divided by 3 through the first pooling layer, in order to reduce the size of the processed data.

C. Analytics strategy

To classify objects in various categories, features are learned by the CNN through its convolutional (conv) layers, which are the heart of the CNN. They are defined by the number of channel it outputs and the conv kernel (weights and bias) of each channel. A conv layer uses one kernel for each channel it has to convolve the input data it recieves. Weights and bias

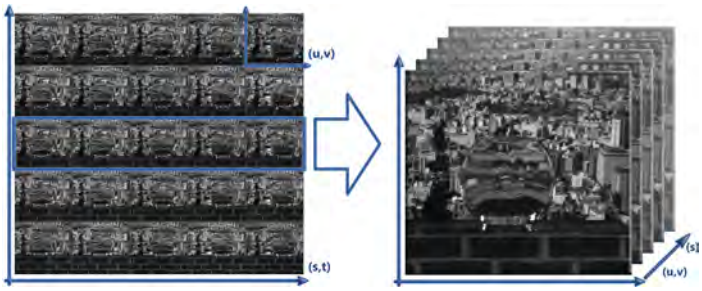

Fig. 1. Selection of frames among LF capture data.

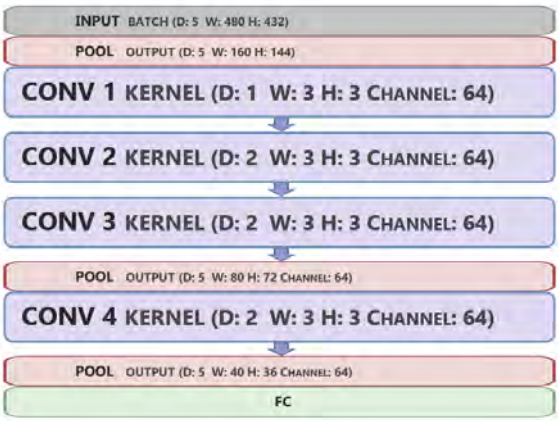

Fig. 2. CNN architecture.

of each kernel are independent from each other, and evolve through the learning process. When the input data is spread among numerous channels, each output channel realize a composition of the input channels that has been convolved by the kernel. Its output ("activation map") is therefore a processed version of the input,and will be transmitted to the next layer of the CNN. The study of each channel can be separated in two parts: the study of a channel characteristics (ie the "features" it uses to process the input data), and its role inside the entire CNN. Studying a channel feature can be done in various ways for a classification process, and this paper focus on two different approaches: The study of a channel when a specific classifier (label) is given, and the study of specific areas in the input data. Since each channel uses the output of numerous channels from the previous layer, a channel may or may not be crucial for the classification process. To tackle this issue, a methodology to identify the importance of a channel in the CNN is also given in this paper. The combination of channel characteristics and importance could allow us to identify what kind of features the crucial channels learn to classify transparent objects in different labels.

III. Compare results from each label

Since our objective is to understand how each channel works, it is crucial to consider every parameter that may vary in our study. For classification, considering each specific object is essential since some channel's feature might be more or less identified within our pannel of labels.

A. Methodology

Once the CNN is fully trained, we can process each label respectively, and get general information to compare with other label's results. Since our dataset is limited, all input data for one label can be combined in a single batch, and the output of each convolutional layer can be compared with other batches. It is also important to remember how the dataset for every object is made: Among the 10 LF pictures of the same object, 8 are used for the training set, and 2 for the validation set. Since for each object the same 8 backgrounds have been used for the training set, it is therefore certified that the only difference between each set of data is the object (for each batch, the same backgrounds are used).

In this study, we consider the sum of the absolute values from each channel's output (its "activation map"), along the the 3 dimensions of the data (Width, Height, Depth).

This CNN is composed of 4 convolutional layers of 64 channels each, in order to classify 20 different types of objects. Since the output of every channel is completely different, we need to normalize them if we want to compare the results for each label more efficiently. This normalization is made by bringing each channel's output between 0 and 1: We focus on the response to each label, rather than the mean difference between each channel.

B. Results

To compare results between label for each channel of the CNN, Fig. 4 presents the normalized sum of activation maps on an intensity scale (if white, the sum has its highest value, if black, it has the lowest). For each layer, channels are randomly generated and evolve through the learning process independently from each other, however some channels' results can have strong similarities between each other. Channels have therefore been reordered by similarities in this publication, and the same rearrangement have been made for label and area processing. Fig. 3 illustrates how results presented in Fig. 4 has been obtained (Details of reordering are not given).

C. Interpretation

For most of the channels, some clear differences can be observed between the labels, and apart from some specific channels (first layer, channel 1 to 25), we obtain very different result between them: Only by observing the most general aspect of activation maps, it is possible to compare how channels percieve each label. More rarely, some channels's response is limited to very few objects (layer 1 channel 59 to 61): with a deeper analysis of the CNN we might observe that this kind of channel is crucial for the detection of these specific labels. Despite the normalization, some channels seems equally active for every label. When checking those channels, their output maps are always at zero: they are not used by the CNN and the final result do not change if they are taken out of the classification process.

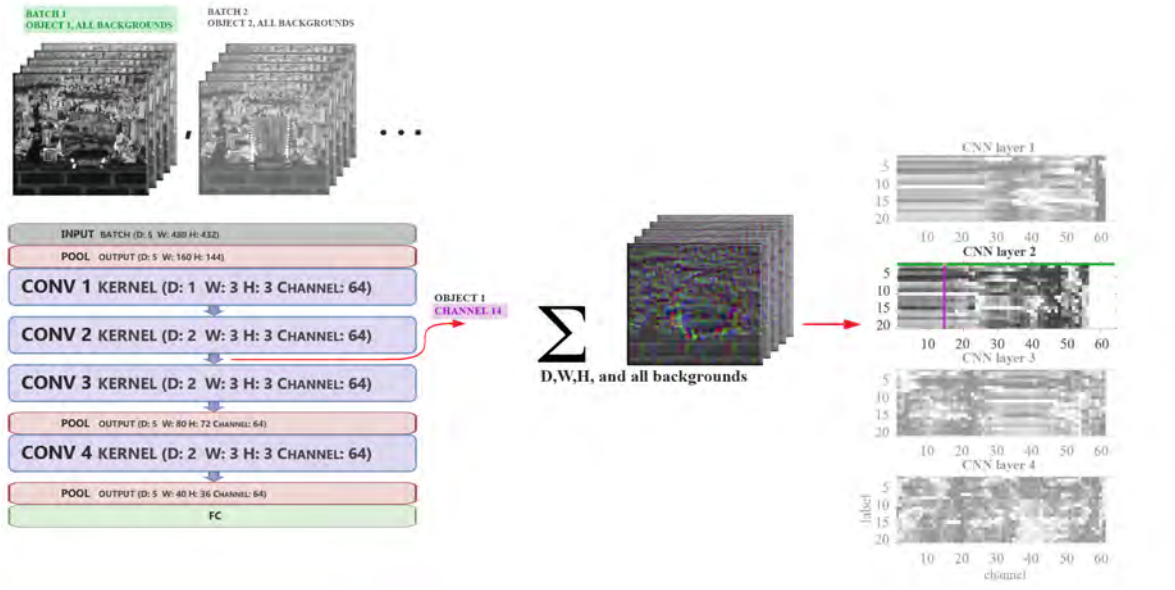

Fig. 3. Each block corresponds to one of the 4 convolutional layer's output. Each pixel value is the normalized sum of the activation maps for one specific channel (horizontal axis) when a selected object's data is inputed (vertical axis).

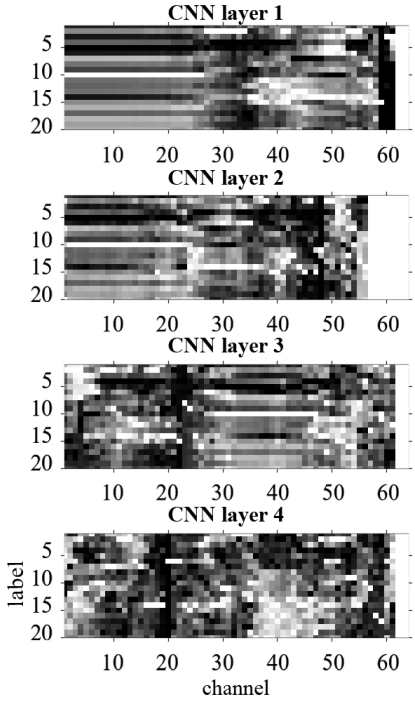

Fig. 4. Normalized sum of each channel's output.

IV. Compare results from each area

Our first approach in identifying each channel's characteristics consist on considering that our data is composed of 3 areas: the background, the edges of the object, and the inside of the object (cf Fig. 5). We therefore identify which area each channel is mostly activated by. Since some features (like LFD) are in a single area, channels focusing on a different area cannot be using this feature.

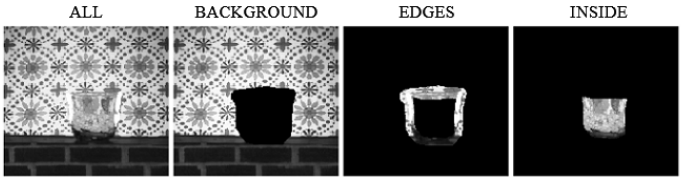

Fig. 5. Input data: Delimitation of each area

A. Areas' evolution through the CNN layers

The areas are being modified by each CNN layer, especially when conv layers imply the "Depth" axis: Between each frame of the depth, the object slightly moves, and areas of both frames must be taken into account.(cf Fig. 6)

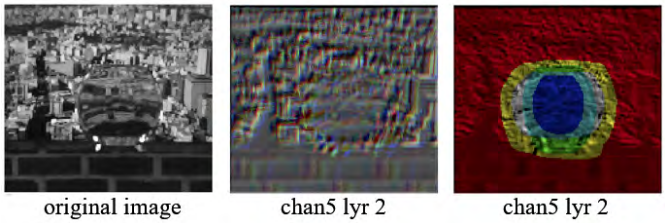

Fig. 6. Output of a specific channel, colored by area. Background area is in red, edge is in green, inside is in blue. When the different area of various depth frames are superposing, a mix of those colors is obtained

B. Comparing activation maps' values for each area

For each channel, we compare the mean of activation values among each different area: The area with the highest value will be the one the channel's feature is focused on. Using a color

for each area, channels for which the feature is mostly reacting to background data will be colored in red, edge-focusing channels will be in green, and inside-focusing channel will be in blue. We apply this method for each element of the batch, since some channels might be more active on specific classifiers.(cf Fig. 7)

C. Results

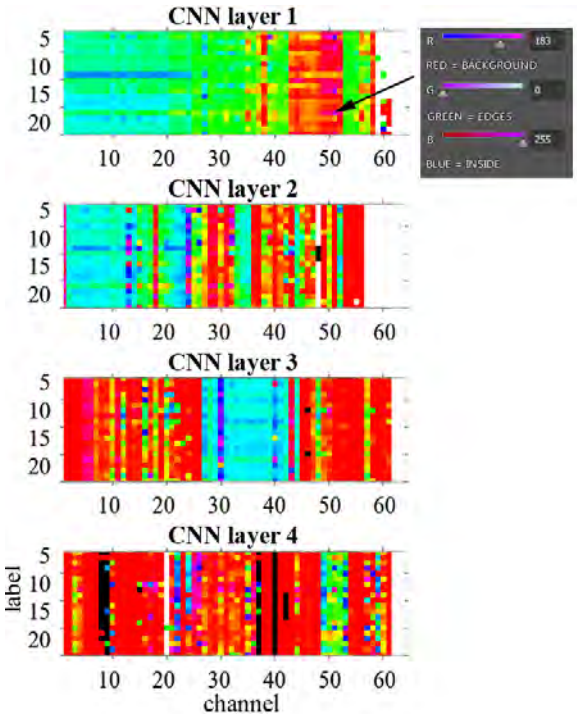

Fig. 7. Each color correspond to the area major area for a specific label, for a given channel.

Most of the results will be displayed in part V/B/2 of this paper since combining results from various methods offer the most interesting analysis. Surprisingly, channels focusing on the background area gets more important when diving through the CNN's layers: analysing the importance of such channels is essential to understand the classification process. After reordering channels the same way than before, we notice the apparition of patterns between the same channels than those on label comparison.

V. How does each channel influence the classification process?

For now, most of our attention has been given to normalized results from each channel. Nevertheless, un-normalized results show that some channels'outputs are way more important than others. Our objective here it to know if mostly ativated channels are more important in the classification process, and what makes a channel important.

A. Killing method

Once the CNN is fully trained, this method consists on setting a chosen channel to 0 ("killing" the channel), and compare the CNN's output with its standard output (when no channel is killed). By doing the same process than Fig. 4, we can efficiently establish which channel of the lower layers use the chosen killed channel, and on which specific labels. By obtaining the confusion matrix of the CNN in every case, we can also know the importance of the channel for the classification process, for each label.

B. Results: Label mapping of each channel

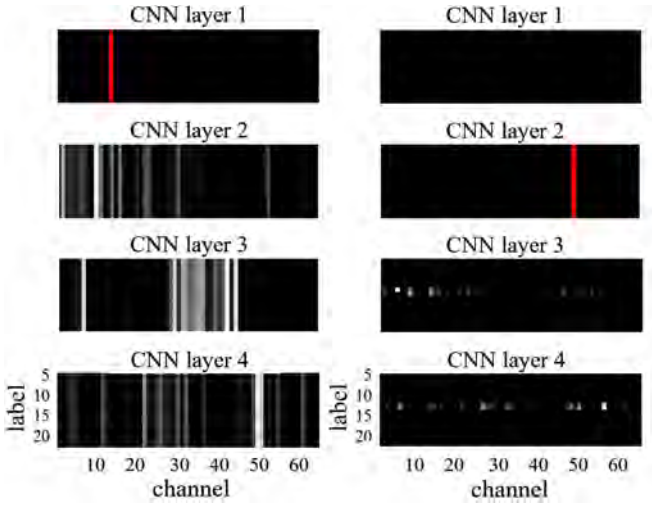

Fig. 8. Killed channel: layer 1, channel 5; then layer 2, channel 22

Fig. 8 contains the difference between standard result (Fig. 4) and result when two different channels have been killed. Entirely black channels are channels that almost do not use the killed channel: As expected, channels from upper layers do not use lower channels, and every channel of the same layer are independent, which is why the first layer's value are at 0 when a channel from the second layer is killed. Some channels also only influence specific labels, which is why we get such different results for the second killed channel. This analysis also confirms that unused channels do not influence the rest of the CNN.

The two previous methods allowed us to identify similar channels (channels that focused on the same labels, and the same areas). Interestingly, those channels are used by the same channels on the lower levels: For such channels, maps as Fig. 8 are different. Two reasons can explain this: either the channels are different on other features than those studied in the previous methods, or the CNN structure is not efficient enought to see they are similar, and they can be replaced by one another. In such a case, we could improve the efficiency of the CNN and save the neccessary time or ressources to process two permutable channels.

C. Results: Confusion matrices

When each column contains the number of estimated label, each column contains the real labels. Therefore, a classification process reaches a perfect score when its confusion matrix is diagonal. For this part, only the training set has been used,

since we get a 100% recognition rate with it. This way, differences can be more easily identified: any non diagonal result would be different from the original results.

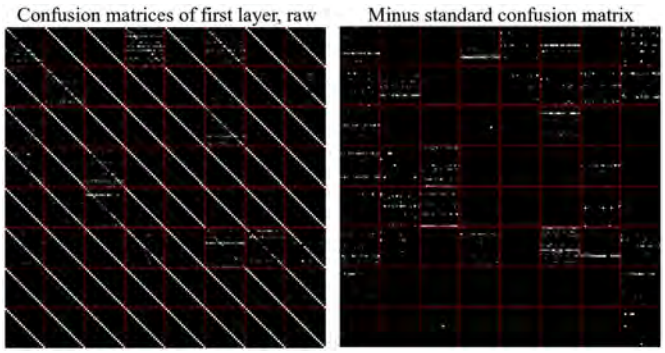

Fig. 9. All 64 confusion matrices when one of the first layer's channel is killed

When outputting the differences only (ie only when the CNN makes mistakes), it becomes clear that, with the training set, a lot of channels could be left with not change of efficiency. Among those channels, we can find those who are unused by the network that we identified previously. Nevertheless, with any validation set, some confusion matrix among those channels might change: therefore these channels would be used for classification, but their role would not be important enought to shift the result with the training set. If some channels are important in various cases, other are specifically useful to identify ("is") or differenciate ("is not") objects. Those channels' confusion matrix have strong values along the columns or lines

VI. Combining results

By combining channel characteristics and importance,we can identify what kind of features the crucial channels learn to classify transparent objects. For example, we can study the final results (confusion matrices) when channels associated to a specific label are killed, or channels associated to a specific area.

A. Modifying channels associated to labels

Among various characteristics that can be identified on the classified label map (Fig. 4), we decided to focus on two: channels focused on specific labels, and channels with similar results.

1) Focused channels: Deleting those channels should give results very different from deleting channels more general: if those channels ensure the object does not belong to a specific category, then the number of objects interpreted as this one should get higher. If it helps confirm the input belongs to a category, the number of object identified as this one should drop.

2) Similar channels: Similar channels tend to focus on identical features, and might therefore be permuted. To confirm this, we tried to copy the same channel in channels similar to this one. Testing reveals that despite their similarities, some

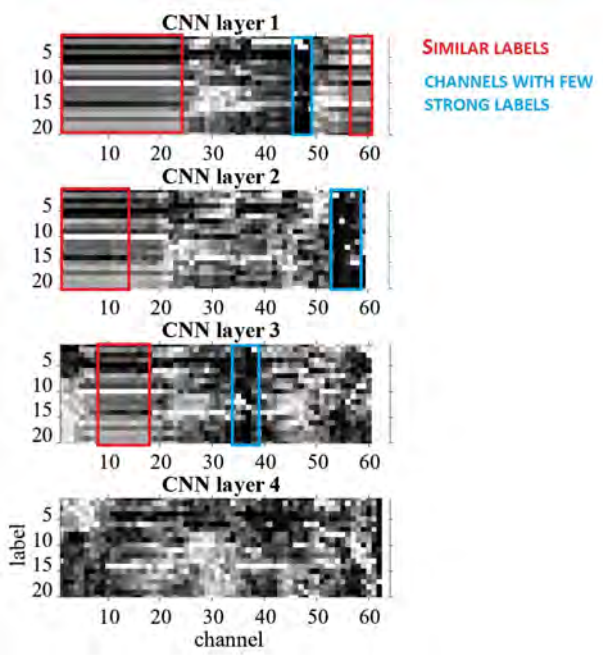

Fig. 10. Area of interest from classified label mapping

of those channels were focusing on different features and therefore could not be permuted (as concluded in section IV/B). However, for some channels, accuracy is almost not impacted: For example, channel 2 of the first conv layer is important for the classification process (when it is set to 0, the accuracy drops significantly, as reveals its confusion matrix on Fig. 11), however when it is replaced by the channel 1 of the first layer, accuracy drops by less than 5% .

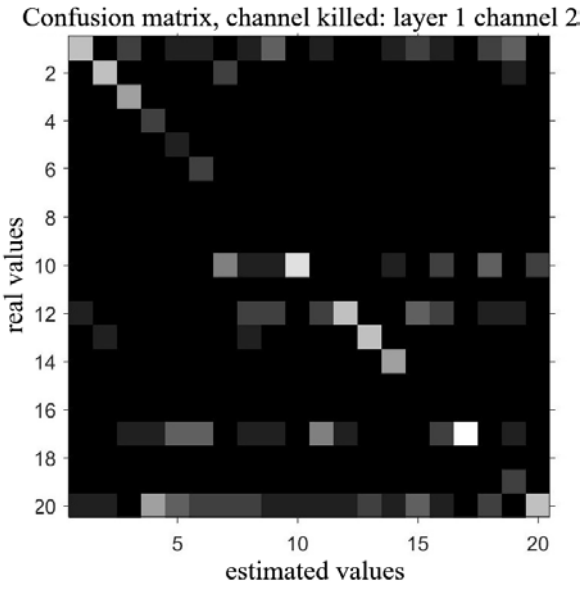

Fig. 11. This confusion matrix reveals how important this channel is: When killed, using the training set, accuracy drops from 100% (diagonal matrix) to 38.75%

Following careful indicators, some channels could be com-

pletely replaced by others with no impact on accuracy and increase of processing speed.

B. Changes according to the focused area

1) Background-focused channels (ie "bgd channels") on layer 1 (11 channels): When killing only one channel at a time, the mean accuracy obtained with bgd channels is 91% on layer 1, but this mean accuracy for any channel on layer 1 is at 84 %. Moreover, the most important of the bgd channel (accuracy of 69 %) is the one that partially focused on the inside area for certain labels. Bgd channels seem to have less impact than others on classification for the first layer.

2) Non bgd channels on layer 3 (15 channels): When killing only one channel at a time, the mean accuracy obtained with those non bgd channels is 44.5% on layer 3, but this mean accuracy for any channel on layer 3 is at 84. %. Moreover, when all non bgd channels are deleted, classification only outputs 2 labels for every input data (label 4 and 9). It therefore seems that, at layer 3, non bgd channels are essential for classification.

3) Non bgd channels on layer 4 (5 channels): On the last layer, the most important channel (accuracy drops to 50% when killed, when it only drops to 86% when the second most important is killed) is associated to edges or inside areas (not background). Accuracy drops to 30% when those five channels are killed at the same time.

Despite having much more channels focused on the background area, the most important channels seem to be associated to other areas. However, bgd channels are also essential for the classification process, since non bgd channels on the deeper levels of the CNN also use bgd channels of higher layers.

C. Discussion

Even though we get an average 82% recognition accuracy at the end of our training with the validation set, this encouraging result might be explained by the very limited dataset we currently use: For example, the number of background realted channels is higher than explected since the object is independent from this area. Although this methods needs refining, it gives interesting results, and its principle can be adapted for other studies (for example, a similar study along the depth of the data can also be worth of interest). Pursuing the study of each specific kernel's perks can allow us to build (automatically or manually) a CNN from pre existing, efficient convolutional kernels, while skipping most of the training part (which is one of the most expensive part when using a CNN). Once perfected, such study might not even need as important dataset as we currently need to build efficient CNN.

VII. Conclusion

This paper presented various CNN analysis methods that has been tested and cross-analysed in order to have a better understanding of the features learned by the CNN to classify transparent objects. Finding out how each channel consider different labels for the classification process, and which area of the input images were mostly activated help us grasping the features identified by each channel: Area analysis can be especially useful in order to identify features associated to a specific area. Combining this analysis with an estimation of each channel's importance allow us to understand how efficient a feature is for the classification process. With this work focused on transparent object classification, two strong conclusions were found:

- Despite being the most essential channels, non background channels on lower levels use (and therefore depends on) background channels, wich are more numerous but less important.
- The CNN can be highly improved: some channels have little to no influence for the classification, but yet are processed. Moreover, some channels learn similar features and could be interchanged: keeping only one of such channels could reduce the processing cost.

By extending this work to other classification process, one can have a better understanding of his CNN (for example, identify which channels are sensitive to high gradient, and see the consequences of blurring the input on such channels and the final result). Moreover, studying the features and importance of each channel can help improving and optimizing a CNN. By automatically improving a CNN along some controlled parameters, a fully trained and optimized CNN could process data faster and more efficiently than before. Identifying which kind of channels are essential to specific classification processes could allow us to initiate a CNN with adequate values and therefore greatly reduce the learning process.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Number JP15K12066, and JST PRESTO Grant Number JPMJPR1505.

References

[1] H.Bay, A.Ess, T.Tuytelaars, L.V.Gool, Speeded-up robust features (SURF), Comput. Vis. Image Underst. 110 (3) (2008) 346359.
[2] Y. Xu, K. Maeno, H. Nagahara, A. Shimada, R. Taniguchi, "Light field distortion feature for transparent object classification, " Kyushu Univ. Fukuoka, February 2015
[3] K. Maeno, H. Nagahara, A. Shimada, R. Taniguchi, Light field distortion feature for transparent object recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 27862793.
[4] D. Miyazaki, K. Ikeuchi, Inverse polarization raytracing: estimating surface shapes of transparent objects, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 910917.
[5] G.S. Settles, Important developments in schlieren and shadowgraph visualization during the last decade, in: International Symposium on Flow Visualization (ISFV),2010.
[6] M. Fritz, M.J. Black, G.R. Bradski, S. Karayev, T. Darrell, An additive latent feature model for transparent object recognition, in: Neural Information Processing Systems (NIPS), 2009
[7] https://www.ptgrey.com/Content/Images/uploaded/KB-Data/ProFUSION_ 25_ datasheet.pdf.
[8] https://github.com/kjchavez/cnn-project