Dynamic Relevance: Vision-Based Focus of Attention using Artificial Neural Networks

Shumeet Baluja
Justsystem Pittsburgh Research Center, 4616 Henry Street, Pittsburgh, PA. 15213 &
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. 15213
baluja@cs.cmu.edu

Dean Pomerleau
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. 15213
pomerleau@cs.cmu.edu

September 11, 1997

Abstract

This paper presents a method for ascertaining the relevance of inputs in vision-based tasks by exploiting temporal coherence and predictability. In contrast to the tasks explored in many previous relevance experiments, the class of tasks examined in this study is one in which relevance is a time-varying function of the previous and current inputs. The method proposed in this paper dynamically allocates relevance to inputs by using expectations of their future values. As a model of the task is learned, the model is simultaneously extended to create task-specific predictions of the future values of inputs. Inputs that are not relevant, and therefore not accounted for in the model, will not be predicted accurately. These inputs can be de-emphasized, and, in turn, a new, improved, model of the task created. The techniques presented in this paper have been successfully applied to the vision-based autonomous control of a land vehicle, vision-based hand tracking in cluttered scenes, and the detection of faults in the plasma-etch step of semiconductor wafers.

1. Introduction

Many real world tasks have the property that only a small fraction of the available inputs are important at any particular time. On some tasks, the extra inputs can easily be ignored. However, often the similarity between the important input features and the irrelevant features is great enough to interfere with task performance. Two common examples of this phenomenon are speech recognition in a noisy environment and image processing of a cluttered scene. In both cases, the extraneous information in the inputs can be easily confused with the important features, making the specific task much more difficult.

Many studies of relevance pre-process the inputs to remove those that are deemed irrelevant [2, 8, 12]. However, in domains such as vision, the smallest set of inputs needed to classify every example (*i.e.*, the union of the sets required for each example) may be the entire set of inputs, while the set needed to classify each individual example may be very small. For example, consider visually tracking a moving object across a large pixel-based input; the goal is to constantly report the value of some attribute of the object, such as orientation. The pixels that are relevant to the task will change as the object moves. Assuming the object appears everywhere in the inputs with equal probability, no pixel can be considered more important than another when the entire training set is considered. Dynamically focusing attention on selected portions of an input scene is a form of allocating relevance to particular inputs at different times.

In this paper, *expectations* are used to guide where to focus attention. Once the important features in the current inputs are found, an expectation of what and where the important features in the next set of inputs can be explicitly created. The next section introduces relevancy/saliency map and describes how these maps are used to create task-specific expectations. Section 3 describes a vision-based autonomous road following system which uses these expectations to de-emphasize unexpected features in the input. Section 4 explores two alternate uses of relevancy

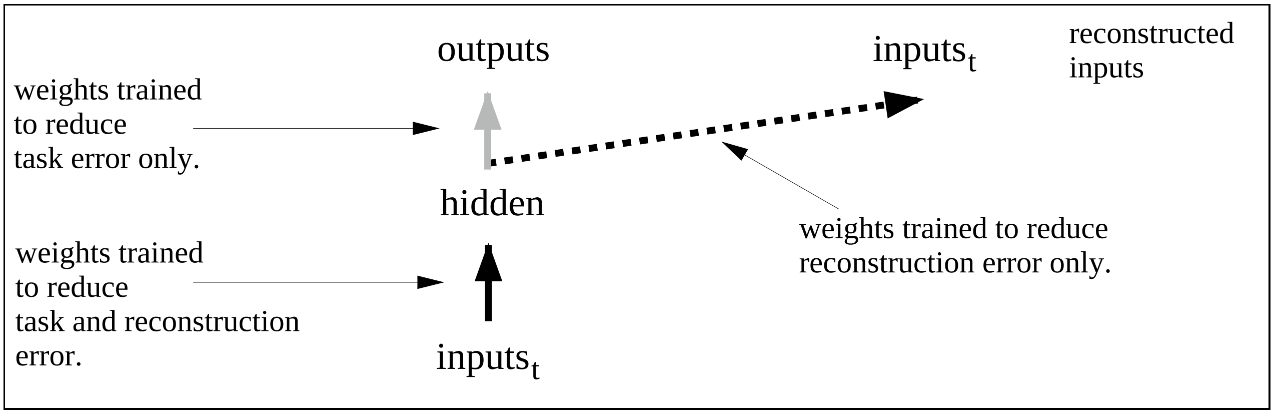
outputs
inputs$_t$
reconstructed inputs
weights trained to reduce task error only.
hidden
weights trained to reduce reconstruction error only.
weights trained to reduce task and reconstruction error.
inputs$_t$

Figure 1: IRRE: using the hidden layer to encode information for reconstruction and task completion.

maps. In contrast to the approach used for road following, the first system reported in Section 4 uses expectations to emphasize the unexpected features in the inputs. The resulting system is used to detect faults in the plasma-etch step of semiconductor wafer fabrication. The second system described in Section 4 is used for hand-tracking. In this system, *a priori* domain-specific knowledge of the task is used to create the expectations of the next inputs. Section 5 and Section 6 discuss conclusions and future directions for research.

2. Relevancy/Saliency Map

In order to direct processing to only the relevant portions of the input, the relevant portions must first be determined. In vision-based tasks, saliency, or relevancy, maps are commonly used tools to indicate the importance of different regions of the input scene. In many studies, saliency maps have been constructed in a bottom-up manner [9, 13]. One method of creating saliency maps within an image is by emphasizing all inputs that differ from their surrounding inputs; this was explored by Koch and Ullman [13]. In another method, multiple different task-specific feature detectors are used to process the input image. Each type of feature detector may contain a weight associated with it, to signify the relative importance of the particular feature. Attention is focused on the regions of the image that contain high weighted sums of the detected features. Top-down knowledge is used to decide which features are used, the weightings of the features, and how many regions are focused upon [1, 9].

Although saliency maps are integral to the techniques presented in this study, the creation and use of them is very different from the procedures described above. In this paper, the representation of an neural network hidden layer, trained to perform a time sequential task, is used to predict what the next inputs will be. In the method proposed here, the expectation of what the features will be in the *next* frame plays a key role in determining which portions of the next visual scene *will be* focused upon. Throughout the rest of the paper, focus of attention and relevancy will be discussed in the context of artificial neural networks (ANNs). In contrast to the top-down method described above, the features and their weightings are developed as the neural network learns to use the features. No *a priori* top-down information is assumed.

2.1. Determining Task-Specific Importance

The method explored in this paper to determine task-specific importance of the inputs is based on *Input Reconstruction Reliability Estimation (IRRE)* [17, 18]. IRRE has been used to estimate the reliability of a network's outputs. The hidden units are used to both reconstruct the input image and complete the main task. The greater the similarity between the actual input image and the reconstructed input image, the more the internal representation has captured the input features, and therefore the more reliable the network's response [18]. Figure 1 provides a schematic of IRRE.

Because the weights between the input and hidden layers are trained to reduce both task and reconstruction error, a potential drawback of IRRE is the use of the hidden layer to encode all of the features in the image, rather than only the ones required for solving the particular task [18]. This can be addressed by noting the following: if a strictly layered (connections are only between adjacent layers) feed-forward neural network can solve a given task,

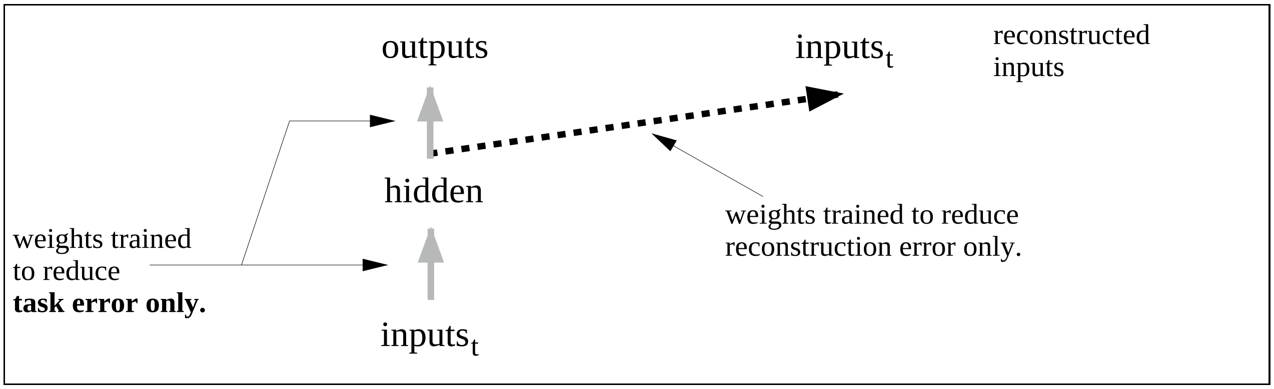
outputs
inputs$_t$
reconstructed inputs
hidden
weights trained to reduce reconstruction error only.
weights trained to reduce **task error only.**
inputs$_t$

Figure 2: Using the activation of the hidden layer to reconstruct the input. The weights between the input and hidden layers are trained to only reduce task error, *not* reconstruction error. Extra hidden layers can be used.

the activations of the hidden layer contain, in some form, the important information for this task from the input layer. One method of determining what is contained in the hidden layer is to attempt to reconstruct the original input image, *based solely upon the representation developed in the hidden layer*. Like IRRE, the input image is reconstructed from the activations of the units in the hidden layer. *Unlike IRRE, the hidden units are not trained to reduce reconstruction error, they are only trained to solve the particular task.* The input reconstruction is done by changing the weights from the network's hidden layer to the reconstruction outputs only. The weights from the inputs to the hidden layer are not affected by the errors in reconstruction. Therefore, the hidden layer is devoted only to solving the task, see Figure 2.

The inputs will not be perfectly reconstructed in the output layer. The weights of the network between the input and hidden layers are not trained to perform reconstruction; they are instead trained to perform the task. The network's allocation of its limited representation capacity at the hidden layer is an indicator of what it deems relevant to the task. *Information that is not relevant to the task will not be encoded in the hidden units.* The reconstruction of the inputs is based solely on the hidden units' activations. Therefore, unlike auto-encoding networks (networks in which the output it trained to reproduce the input layer [3, 14]) and principal components analysis, the irrelevant portions of the input are not encoded in the hidden units' activations, and the inputs that are irrelevant to the task *cannot* be reconstructed[1] [4, 7].

A notion of time is necessary in order to focus attention in *future* frames. Instead of attempting to reconstruct the current input, the network is trained to predict the next input (this corresponds to changing the subscript of the reconstructed inputs in Figure 2 from “t” to “t+1”). The prediction is trained in a supervised manner, by using the next frame in the time sequence as the target. The target (the next inputs) may contain noise or extraneous features. However, since the hidden units are only intended to encode information to solve the task, the network will be unable to construct the noise or extraneous features in its prediction. This is expanded upon in the next section.

2.2. Removing Irrelevant/Distracting Features

The first application explored in this paper is autonomous road following, in particular lane marking detection. For this task, extraneous features should be removed and the lane marking emphasized. Therefore, expectation will be used to remove the unexpected inputs.

In many vision tasks, filtering can be done at a very low level, for example on a pixel-by-pixel basis. In this system, which uses images as input, a saliency map is created by using the absolute differences between the expectation of input image$_{t+1}$ (derived from input image$_t$) and the actual input image$_{t+1}$. This “difference image” is scaled to the range of 0.0 to 1.0, with smaller differences closer to 1.0. The result is the saliency map. To de-emphasize an input pixel, its value is adjusted towards the background value of the image. Since the background may not be uniform across the image, the background value of each pixel is estimated by individually averaging the values of each pixel

[1] The features that can be reconstructed are highly correlated with the important features. The exception to this are inputs that have constant values; these can always be reconstructed. However, these can be removed by simple pre-processing.

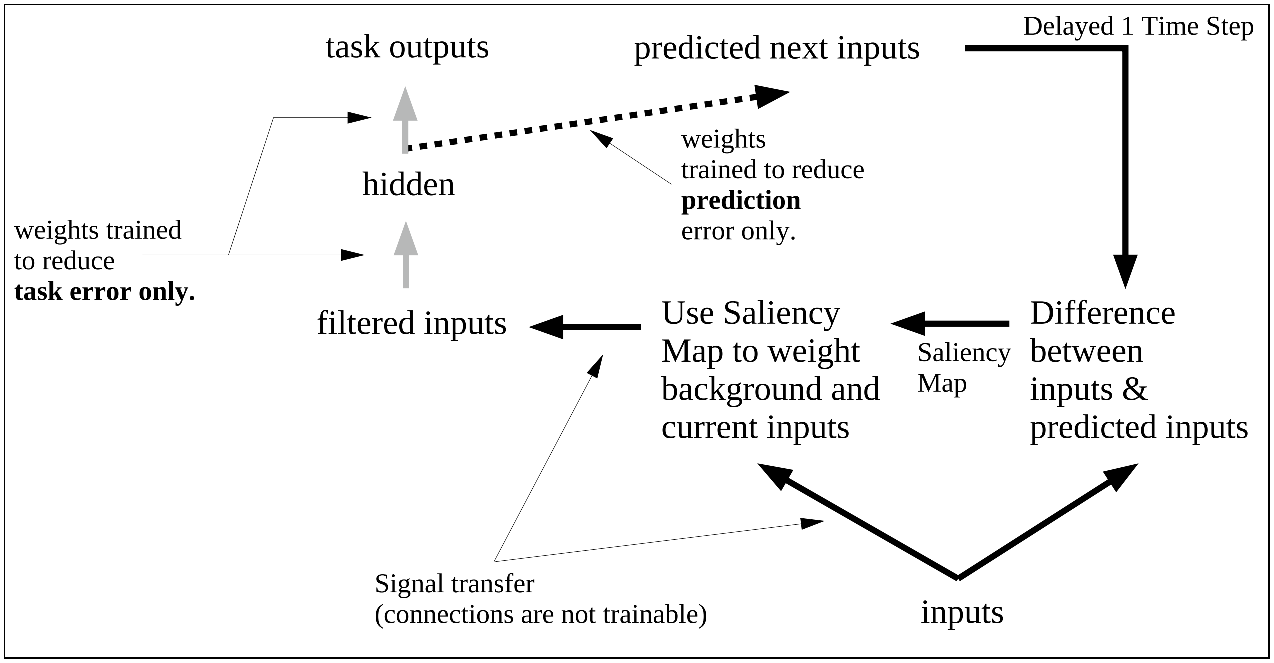

Figure 3: Prediction of next inputs, with feedback. The difference between the predicted and actual inputs is the basis for the saliency map. The background is estimated directly from the training set. Bold arrows indicate signal transfer connections; these are not trainable.

across all of the training examples[2]. Each input to the network is a weighted average of the pixel's background value and current value. The more salient the pixel is, according to the saliency map, the higher the weight on the current value. Therefore, the inputs that match the expected value are left unaltered. The inputs that are very different from the expected value are changed to their background value. The system architecture is shown in Figure 3. It is used in this exact form for autonomous road following, as described in Section 3.

With feedback to the inputs, the system becomes harder to train since the training patterns constantly change. As the model for the task improves, the saliency map becomes more refined, and more of the correct information is given to the network in the next time-step. Therefore, the images input to the network later in the training process possess different qualities than those input earlier in training. Because the network is trained to reduce the task error, the hidden representation changes to adapt to the new images. This changes the prediction of the next inputs, and the cycle continues. In all of the experiments conducted, the dynamics of the system did not prevent convergence. The systems converged using the standard error-backpropagation learning algorithm with small learning rates. The "chicken-and-egg" problem, of needing to determine the features that will be important for solving the task before the task is solved, is avoided in many situations because some of the images may contain no distractions and others will not contain the same types of distractions. Therefore, a small amount of learning is able to proceed without explicit focus of attention. Once a few of the important features are determined, the system bootstraps itself.

3. Lane Marker Tracking

In this section, we briefly summarize the empirical results with an autonomous road following system which uses expectation to dynamically assess the relevancy of the inputs [4, 7]. The goal of autonomous road following is to control a robot vehicle by analyzing the image of the road ahead. The direction of travel should be chosen based on the location of features like lane markings and road edges. This is a difficult task since the scene is often cluttered with extraneous features such as other vehicles, pedestrians, trees, road signs and other objects that can appear on or around a roadway. For the general task of autonomous navigation, these extra features are extremely important. However, for the restricted task of road following, these features can be distracting.

[2] In the implementation described here, only first order information is used to compute the background image.

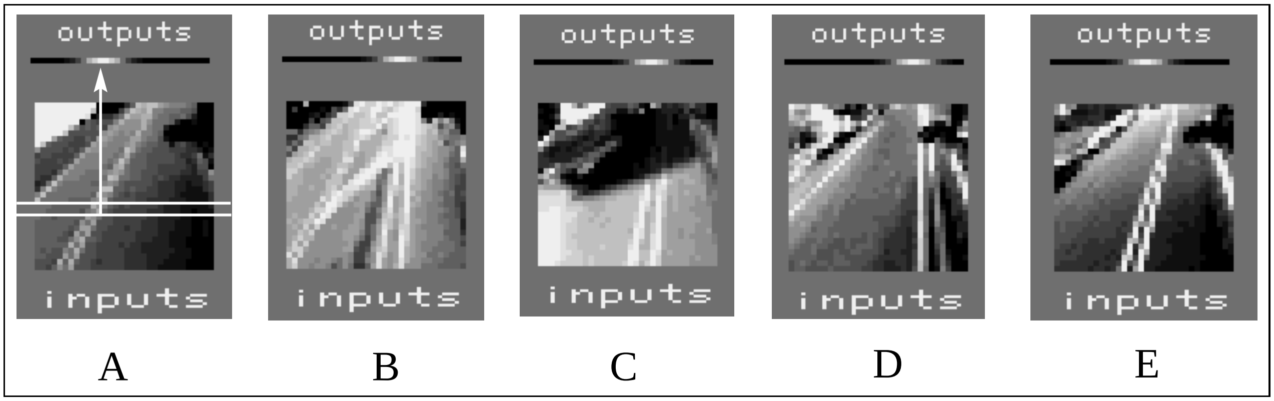

Figure 4: Five sample input images and target outputs. Image A shows the region from which the lane marking was manually selected.

3.1. The ALVINN Road Following System

ALVINN (**A**utonomous **L**and **V**ehicle in a **N**eural **N**etwork) is an artificial neural network based perception system that learns to control CMU's NAVLAB vehicles by watching a person drive [17]. ALVINN's architecture consists of a single hidden layer backpropagation network. The input layer of the network is a 30x32 unit two-dimensional "retina" that receives input from the vehicle's video camera. The correct steering direction is determined from the activation of 30 output units. The output units attempt to create a Gaussian centered around the correct steering direction. If the Gaussian is located around unit 1, this indicates the vehicle should make a sharp left, if the Gaussian is around unit 30, the vehicle should make a sharp right, etc. To teach the network to steer, ALVINN is shown video images from an onboard camera as a person drives. For each image, it is trained to output the steering direction in which the person is steering. Other network architectures, such as radial basis function networks, have also been studied for this task [19]. However, the performance of these simple methods degrade when presented with cluttered environments like those encountered when driving in heavy traffic, or on city streets. In particular, distractions, such as extraneous lane marking or passing cars, may produce incorrect results.

In one of the proposed applications of this system, ALVINN will warn drivers if they begin to drift over lane markings (indicating that they may be entering a lane with on-coming traffic, or leaving the road, etc.). The purpose of using a saliency map is to eliminate features of the road that the neural network may mistake as lane markings. In this experiment, approximately 1200 images were gathered from a camera mounted on the left side of a car, pointed downwards and slightly ahead of the vehicle. The images were gathered sequentially, at approximately 4-5 images per second. The car was driven through city and residential neighborhoods around Pittsburgh, PA. The images were subsampled to 30x32 pixels. From these images, it is possible to steer the vehicle. However, in order to quantify the results with and without the use of the saliency map, an alternate, closely related task was chosen. In each of these images, the horizontal position of the lane marking in the 20th row of the input image was manually identified. The task is to produce a Gaussian of activation in the outputs that is centered around the horizontal position of the lane marking in the 20th row of the image. Sample images and target outputs are shown in Figure 4. In this task, it is vital to focus on only the relevant portions of the input image. If the entire image is used, the artificial neural network can become confused by road edges, as shown in Figure 4a; by extraneous lane markings, as shown in Figure 4b; passing cars, as shown in Figure 4c; and reflections on the car itself, as shown in Figure 4d and 4e.

Assuming that the driver has directed the car well, the center line has probably stayed within a small region of the input image. Therefore, the network has not been trained to recognize lane markings outside the middle regions of the image. To augment the training set, extra images were created by translating the original images to the left or right by up to 5 pixels. The output was also translated either to the left or right by the same amount as the image. This translation yields usable images because the camera is pointed downwards. If the camera had been pointed more ahead of the vehicle, more sophisticated rotations would have been required to maintain the correct perspective [16]. Further details on image creation and insertion into the training set can be found in Baluja's thesis [4].

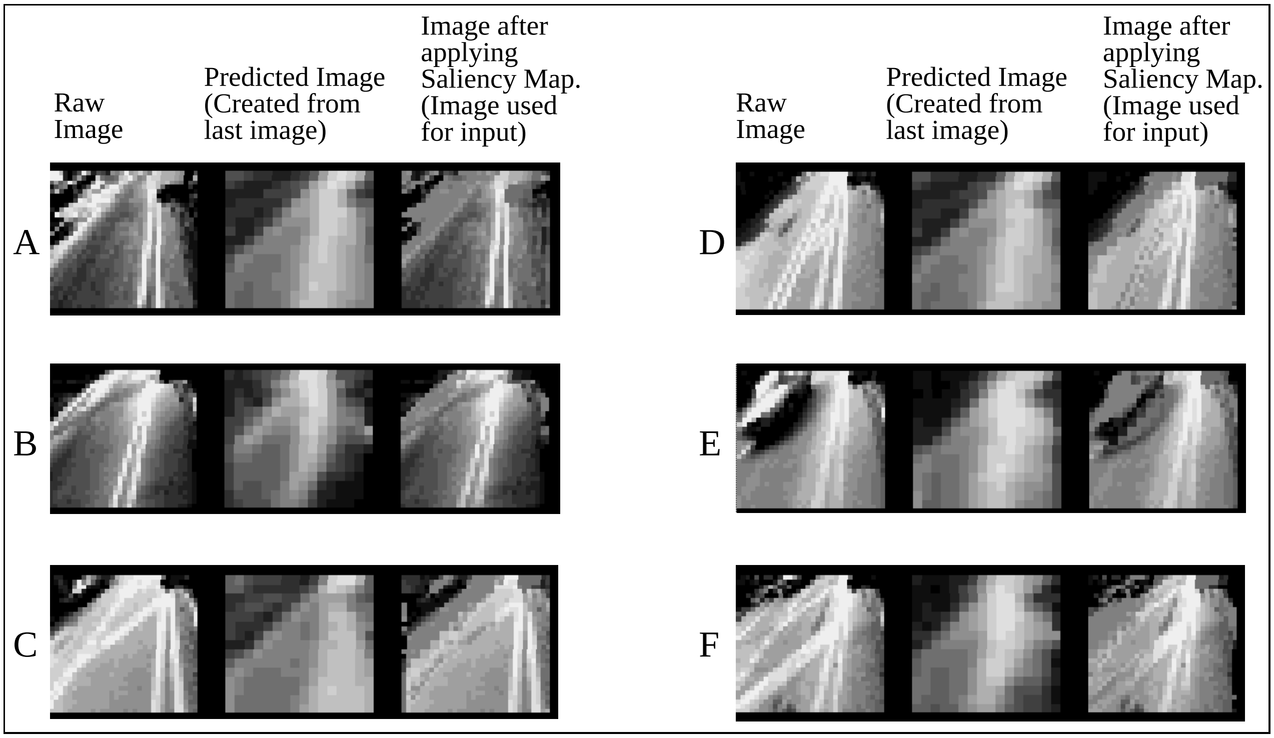
Raw Image
Predicted Image (Created from last image)
Image after applying Saliency Map. (Image used for input)
A
B
C
D
E
F

Figure 5: Left: raw input image$_t$. Middle: the network's prediction of the inputs at time t. This prediction was made by a network with input of image $_{t-1}$. Right: image that is filtered, pixel-by-pixel, by the saliency map; this image is used as the input to the neural network.

3.2. Results

A system that implemented the procedure described in Section 2.2 was trained on the real and synthetic images (created by the procedure described above). The saliency map successfully removed distracting noise from the test images. Figure 5 shows 6 image triplets (the images are scaled between value of -1 and +1). In each triplet, on the left, the original, unfiltered, input image$_t$ is shown. The middle image is the predicted image$_t$, made by the network with input of image$_{t-1}$. The right image has been filtered, pixel-by-pixel, by the absolute difference between the original image$_t$ and predicted image$_t$ (the saliency map). The larger the difference between actual and predicted image, the more the pixel's activation is reduced towards 0 [3]. In triplets A & B, the edge of the road is bright enough to cause distractions. In C & D, the two lane markings may confuse the lane tracker, and cause oscillations between the markings. In E, a passing car creates confusing bright pixels in the image.

In each of these triplets, the confusing attributes are removed from the image (shown in the right image of each triplet). Triplet F shows the limitations of the saliency map; although it is able to eliminate a large portion of the noise, not all of it can be removed. Nonetheless, the most distracting lane marking was removed, and the ANN with the saliency map was not confused by the pixels that were not removed. The expectations displayed in column 2 of Figure 5 show that the expectation of where to find the lane marking and the road edges is not precisely defined. This is due to the training method, that attempts to account for the many possible transitions from one time step to the next by expanding the "expected area" to be more inclusive. It is vital to ensure that the important features do not fall outside of the expected area. Features outside of the expected area may be de-emphasized in processing.

The performance of the lane-tracker with the saliency map, measured by the absolute difference between the predicted location of the lane marking and the hand-labeled position of the lane marking, revealed an average improvement of approximately 20% over the lane-tracker without the saliency map. Without the saliency map, the peak

[3] As described in Section 2.2, we can move the pixel's activation towards the background activation to de-emphasize the inputs. In this implementation, the activation of each pixel is moved towards 0 (the average activation of the pixels that correspond to the road), since we only need to eliminate distractions on the road.

of the Gaussian in the network's output units was an average of 1.15 units away from the hand-labeled peak (as described in Section 3.1, the network has 30 output units). With the saliency map, the network's peak output was an average of 0.92 units away from the hand-labeled peak. The average improvement was not greater because many of the images in the test set do not contain noise; with these noise-free images, a standard ANN (without a saliency map) can be used to accurately estimate the lane marking position.

There are important differences between this application of relevance detection and many of the other studies conducted in non-visual domains. First, the inputs in this domain are highly redundant. Therefore, the filtering does not have to be perfect. If some important pixels are accidentally removed, there should be many other pixels that contain enough information to recover. Second, it is important to eliminate as many of the unnecessary pixels as possible, since they may contain information that directly conflicts with the correct information. Third, the dynamic nature of relevance is exemplified in this domain; as the images change, the relevant pixels change.

It should be noted that exactly the same selective attention mechanisms used for lane marker tracking can be used when the regions of interest are spatially discontinuous over consecutive time-steps. There is no prerequisite for smooth transitions of focusing regions, as is commonly assumed in many vision systems [4, 6].

4. Extending the Use of Relevancy/Saliency Maps

This section briefly describes two alternate uses of the saliency map. First, the saliency map's use in anomaly detection is described. It has successfully been used to detect faults in the plasma-etch step of semiconductor wafer fabrication. Second, methods for incorporating *a priori* relevance information are presented.

4.1. Using the Saliency Map for Anomaly Detection

In the previous sections, the difference between the expected and actual inputs was considered noise, and was deemphasized from processing. However, for anomaly detection, the opposite behavior is desired. *The differences between the expected and the actual inputs are the points of interest*, because they are the regions that were *not* expected. Therefore, they may be anomalies that need to be detected. This interpretation of expectation has applications in the analysis of visual scenes in which the object of interest is moving across a stationary background and for fault/anomaly detection in time-series data.

A real-world problem that benefits from this use of expectation is the detection of faults in the plasma-etch step of the fabrication of semiconductor wafers. For this process, anomaly detection is done by monitoring sensors inside the etch chamber [15]. A typical sensor used for this process produces a waveform representing the intensity of light emitted at various wavelengths during etching. Anomaly detection is complicated because, as wafers are etched, the etch chamber becomes contaminated and the etcher parts degrade. Therefore, the sensor's outputs for the wafers, even those with no-faults, changes over time. Successful fault detection methods must be able to account for these effects, and adapt to a changing underlying process.

To test a system based on the techniques described here, experiments were conducted with a real data set of approximately 4100 plasma-etches [4, 5]. Both neural and non-neural techniques were applied to this problem. In the expectation-based system, an ANN was used to predict the waveform of the next wafer to be etched, given the waveform of the current wafer. Rather than emphasizing the similarities between the predicted and actual waveform (as was done in the autonomous driving domain), *the differences between the predicted and actual waveforms were emphasized*. The difference-waveform, as well as the original and predicted waveforms, were used as inputs into a classification network. Many other systems were also tested on this problem, including single-hidden layer neural networks which made classifications based solely on the current waveform, neural network architectures which incorporated information about the etcher's state by comparing each waveform with the waveform of the last previously found no-fault wafer, and methods based on the magnitudes of the differences between the waveforms of previously found good wafers and the current wafer. Of the systems examined, the expectation-based system worked the best; the missed fault rate dropped from the next best system by almost an order of magnitude, while maintaining a very low false alarm rate [4, 5].

4.2. Incorporating *a priori* Relevance Information

The saliency map can be used as a tool for interacting with external knowledge sources. In the tasks described in this paper, the transition rules were *learned* by the ANN. However, if the relevance transition rules had been known *a* *priori*, processing could have been directed to only the relevant regions by explicitly manipulating the saliency map.

The ability to incorporate *a priori* rules is important in many vision-based tasks. Often the constraints about the environment in which the tracking/detection is done can be used to limit the portions of the input scene that need to be processed. For example, consider tracking a person's hand, for the purpose of creating a gesture recognition/hand tracking system. Given a fast camera sampling rate, the person's hand in the current frame will be close to where it appeared in the previous frame. Although a network can learn this constraint by developing a saliency map based on future input prediction, training can be avoided by incorporating this rule directly.

A system that tracks a user's hand was developed [4]. Rather than creating a saliency map based upon the difference between the actual and predicted inputs, as was done with autonomous road following, the saliency map was explicitly created with the available domain knowledge. Given the sampling rate of the camera and the size of the hand in the image, the salient region for the next time-step was a circular region centered on the estimated location of the hand in the previous image. The activations of the inputs outside of the salient region were shifted towards the background image. The activations inside the salient region were not modified. After applying the saliency map to the inputs, the filtered inputs were fed into the neural network. The system was tested in a typical office setting. To make the tests challenging, a subject, who was not used during training, was asked to wave both hands in the air while opening and closing them at random intervals. The subject's hands and body moved throughout the image sequence. The system was able to track both hands, whether open or closed, in scenes containing clutter and other moving objects, including other hands and other people. In contrast to a system that did not use these attention mechanisms, this system reduced the error by over 80%, measured by the number of frames each system was successfully able to find each hand to within a pre-specified number of pixels[4]. In Figure 6, the results of tracking both the left and right hand in a sequence of 283 images taken from the test sequence described above are presented. The X-axis represents the weight given to the image which has been filtered by the saliency map. At 0.0, no weight is given to the filtered image; therefore, only the raw input images are used. At 1.0, only the image filtered by the saliency map is used; at intermediate values, a weighted combination of the raw and filtered images are used. The results show that using the saliency map significantly improves the performance in tracking both hands. Without the saliency map, the left hand is found more often than the right since it appeared brighter in this image sequence (the left hand was closer to a light source). In general, if symbolic rules exist that provide information on the importance of the input units, a saliency map provides a mechanism to incorporate this knowledge into the neural network.

5. Conclusions

This paper has presented a method for dynamic relevance assessment using artificial neural networks. The network's allocation of its limited representation capacity at the hidden layer is an indicator of what inputs it deems relevant to the task. By using only the hidden layer to predict the next input image, and comparing this prediction with the actual next image, it is possible to estimate which inputs should be ignored and which should be attended. In deciding whether this approach is suitable to a new problem, there are two main criteria that must be considered. First, if expectation is to be used to remove distractions from the inputs, then given the current inputs, the activations of the relevant inputs in the next time step must be predictable. Additionally, the irrelevant inputs must either be unrelated to the task or be unpredictable. In many visual object tracking problems, the relevant inputs are often predictable while the distractions are not. In the cases in which the distractions are predictable, if they are unrelated to the main task, the model will not learn how to predict them. When using expectation to emphasize unexpected or potentially anomalous features, the activations of the relevant inputs should be unpredictable while the irrelevant ones are predictable. This is often the case for anomaly/fault detection tasks. Second, when expectations are used as a filter, it is necessary to explicitly define the role of the expected features. In particular, it is necessary to define whether the expected features should be considered relevant or irrelevant, and therefore, whether they should be emphasized or de-emphasized, respectively.

The most closely related procedure to the one described in this paper is the use of Kalman Filters to predict the locations of objects of interest in the input retina. A successful application of Kalman Filters to focusing attention is described by Dickmanns [10]. Dickmanns uses the prediction of the future state to help guide attention, for example, by controlling the direction of a camera to acquire accurate position of landmarks. Very strong models of the vehicle motion, the appearance of objects of interest (such as the road, road-signs, and other vehicles), and the motion of these objects are encoded in the system. Additionally, a model of the system noise must also be created. There are many

[4] A hand was considered successfully found if the network estimated its position within 7 pixels of the actual position in both the X and Y axis. The total size of the input to the network was 48x48 pixels.

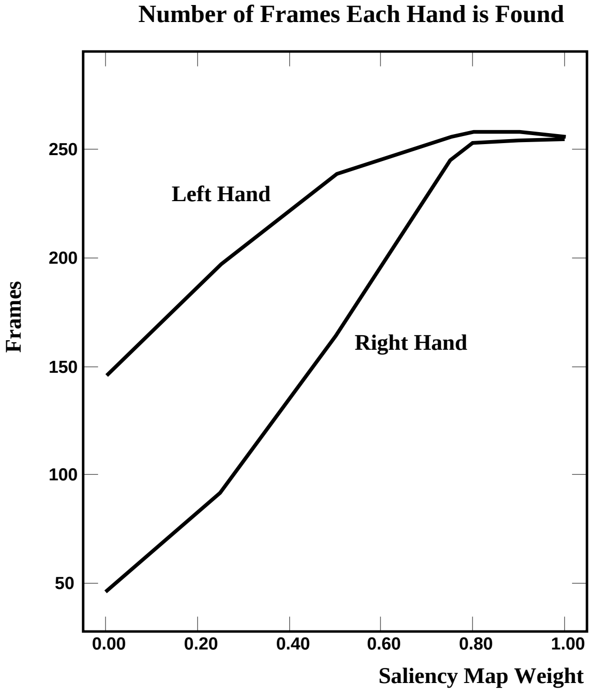

Figure 6: Results for hand-tracking experiments. The X axis is the weight of the filtered image compared to the unfiltered image. The Y axis is the number of frames in which each hand was found. There are a total of 283 images in the sequence.

similarities between this system and ones that employs Kalman Filters [4]; however, there are also very important differences. The largest difference is the amount of *a priori* knowledge that is used. Many approaches that use Kalman Filters require a large amount of problem specific information for creating the models. In the approach presented in this paper, the main object is to automatically learn this information from examples. First, the system must learn what the important features are, since no top-down information is assumed. In many tasks this is a difficult problem. For example, in lane marker tracking problem discussed in Section 3, there are several features that should be modeled. Because of shadows of passing cars, it is insufficient to only model the lane marking's transitions in successive frames. The networks developed in our study also model the road edges, since these are vital to determining the lane marker location when the lane marker is not visible. Second, the system must automatically develop the control strategy from the detected features; this is the standard task of neural networks. Third, the system must also generate a model for the movements of all of the relevant features; this is the prediction portion of the network.

The first task, autonomous road following, was chosen as a test-bed for the attention mechanisms since it is representative of many real-world tasks that require the real-time analysis of large amounts of incoming sensor data. This task also serves to illustrate the dynamic nature of relevance. The ANN used in this study was able to avoid distractions by focusing attention on only the relevant portions of a scene. In the task of autonomous road following, the algorithm is not misled by extra lane markings and other features that have similar appearances. Second, the use of expectations in a system which detected faults in the plasma etch step of semiconductor wafer fabrication demonstrated that expectations can be used for anomaly detection by emphasizing surprise features. Third, the task of hand tracking in cluttered scenes was explored to show that when *a priori* information is available, it can easily be incorporated into the attention mechanisms.

6. Future Directions

There are many directions for future research. This section describes three that are of immediate interest. For dynamic relevance assessment, this paper has presented empirical evidence to support the use of expectations and information gained from task-predictability. In the future, we hope to formally analyze simplified portions of the networks used in this paper. For example, the non-linearities introduced both in the neural network and in the various filtering mechanisms can be replaced with simpler functions that are more amenable to analysis.

The systems described have used only a single previous time-step to make predictions of the next inputs. However, this procedure is not limited to Markovian decisions. There are many methods of incorporating more state information. For example, in the lane-marker tracking task, instead of only using the current image to predict the next image, the current actions, or multiple previous actions, could have been used. Additionally, multiple previous images can be employed for making predictions.

Ideally, in visual domains, filtering should be done at the *object* level. However, this type of filtering requires object detection procedures to analyze the scene and find the relevant objects. Unfortunately, object detection procedures are themselves complicated and are topics of ongoing research [11, 20, 21]. To avoid the need for object detection, the input-pixels are filtered based on their expected values. An extension of expectation-based filtering is to filter the *hidden* layer of the trained neural network instead of the inputs. Rather than predicting the activations of the next inputs, the next time step's activations of the hidden layer would be predicted. By moving the filtering to the hidden layer, processing is directed away from the pixels and towards higher-level features [4].

Acknowledgements

The authors would like to thank Tom Mitchell, Takeo Kanade and Tomaso Poggio for their help in shaping this work. The authors would also like to thank Roy Maxion for kindly providing the data set for the semiconductor fabrication experiments. Shumeet Baluja was supported by a National Aeronautics and Space Administration Graduate Student Fellowship, administered by the Lyndon B. Johnson Space Center. This work was started while Shumeet Baluja was supported by a National Science Foundation Graduate Fellowship. Additional support has come from "Perception for Outdoor Navigation" (contract number DACA76-89-C-0014, monitored by the US Army Topographic Engineering Center) and "Unmanned Ground Vehicle System" (contract number DAAE07-90-C-R059, monitored by TACOM). Support has also come from the National Highway Traffic Safety Administration under contract number DTNH22-93-C-07023. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of NASA, NSF, ARPA, or the U.S. Government.

References

[1] S. Ahmad. *VISIT: An Efficient Computational Model of Human Attention*. PhD thesis, University of Illinois at Urbana Champaign, 1991.

[2] H. Almuallim and T. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth International Conference on Artificial Intelligence*, pages 547–552, San Jose, CA, 1991. AAAI Press.

[3] P. Baldi and K. Hornik. Neural networks and principal components analysis: Learning from examples without local minima. *Neural Networks*, 2:53–58, 1989.

[4] S. Baluja. *Expectation-Based Selective Attention*. PhD thesis, Computer Science, Carnegie Mellon University, Pittsburgh, PA, September 1996. Available as CMU-CS-96-182.

[5] S. Baluja and R. Maxion. Artificial neural network based detection and diagnosis of plasma-etch anomalies. *Journal of Intelligent Systems*, 7(1-2):57–82, 1997.

[6] S. Baluja and D. Pomerleau. Using a saliency map for active spatial selective attention: Implementation and initial results. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 451–458, MA, 1995. MIT Press.

[7] S. Baluja and D. Pomerleau. Expectation-based selective attention for visual monitoring and control of a robot vehicle. *Robotics and Autonomous Systems Journal*, To appear, 1997.

[8] R. Caruana and D. Freitag. Greedy attribute selection. In W. Cohen and H. Hirsh, editors, *Proceedings of the Eleventh International Conference on Machine Learning*, pages 28–36, San Mateo, CA., 1994. Morgan Kaufmann Publishers.

[9] J. Clark and N. Ferrier. Attentive visual servoing. In A. Blake and A. Yuille, editors, *Active Vision*, pages 137–154, MA, 1992. MIT Press.

[10] E. Dickmanns. Expectation-based dynamic scene understanding. In A. Blake and A. Yuille, editors, *Active Vision*, MA, 1992. MIT Press.

[11] T. Huang and S. Russell. Object identification in a bayesian context. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, San Jose, CA, 1997. AAAI Press.

[12] G. John, R. Kohavi, and K. Pfelger. Irrelevant features and the subset selection problem. In W. Cohen and H. Hirsh, editors, *Proceedings of the Eleventh International Conference on Machine Learning*, pages 121–129, San Mateo, CA., 1994. Morgan Kaufmann Publishers.

[13] C. Koch and S. Ullman. Selecting one among the many: A simple network implementing shifts in selective visual attention. Technical Report MIT-AI Memo 770, MIT, MA, 1984.

[14] M. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–242, 1991.

[15] R. Maxion, 1995. Semiconductor Wafer Plasma-Etch Data Set, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

[16] D. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

[17] D. Pomerleau. *Neural Network Perception for Mobile Robot Guidance*. Kluwer Academic Publishing, Boston, MA, 1993.

[18] D. Pomerleau. Reliability estimation for neural network based autonomous driving. *Robotics and Autonomous Systems Journal*, 12:113–119, 1994.

[19] M. Rosenblum and L. Davis. The use of a radial bases function network for visual autonomous road following. Technical Report Technical Report CAR-TR-666, University of Maryland Center for Automation Research, 1993.

[20] H. Rowley, S. Baluja, and T. Kanade. Neural network based face detection. In *Proceedings of Computer Vision and Pattern Recognition 1996*, pages 203–208. IEEE Computer Society, 1996.

[21] K.K. Sung. *Learning and Example Selection for Object and Pattern Detection*. PhD thesis, MIT - AI Lab, 1996. Technical Report 1572.