CARLO CILIBERTO: Good morning. So today, there is a bit if a overview on the iCub Robot, and it will be about one hour, one hour and half. And we organized the schedule for this time in series of four small talks and then a demo from the iCub, live demo. So I will give you an overview of the kind of fields and capabilities that the iCub has developed so far, while Alessandro, Rafaello, and Giulia will show you what's going on right now on the iCub, part of what's going on the robot. So they are going to talk about their recent work.

So let's start with the presentation. This is the iCub, which is a child humanoid robot. This size. And the project of the iCub began in 2004, and the iCub-- actually, the iCubs, because there are many of them-- they were built in Genoa at the Italian Institute of Technology. And the main motivation behind the creation and the design of this platform was to actually have a way to have a platform in order to study how intelligence, how cognition emerges in artificial embodied systems.

So Giulio Sandini and Giorgio Metta, that you can see there, are the actual original founders of iCub world, so they are both directors at the IIT. And this is a bit of a timeline that I drew. Actually, there are many other things going on during all these 11 years. This is a video celebrating the actual first 10 years of the project. And actually, you can see many more things that the iCub will be able to do. But I selected this part because I think they can be useful, also, if you are interested in doing projects with the robot, to have an idea of the kind of skills and the kind of feedback that the robot can provide you, to do an experiment.

So as I told you, iCub was built with the idea of replicating an artificial embodied system that could explore the environment and learn from it. So it has many different sensors. These are some of them. So in the head, we have one accelerometer and the gyroscope in order to provide the inertial feedback to the system, and two Dragonfly cameras that provide medium resolution images. And as you can see there, it's all about one meter, one meter and something, and it's pretty light-- 55 kilograms-- and has a lot of degrees of freedom. So 53 degrees of freedom, and they allow the robot to perform many complicated actions.

It's provided with torque and force sensors, and I will go over these in a minute. Its whole body, or at least the covered part of the robot-- the black part that you can see there-- are all covered in artificial skin. So they provide feedback about contact with the external world.

And it has also microphones mounted on the head, but probably for sound and speech recognition is better to use direct microfeedback at the moment, because, of course, noise-canceling problems and so on. If you're interested in speech and found feedback, we are going to use other kind of microphone.

So during these 11 years, iCub has been involved in many, many projects, and indeed, part of what I'm going to show you is the result of the joint effort of many labs, mainly in Europe. These are mostly European projects. But iCub is also an international partner of the CBMM Project.

So regarding force/torque sensors, they are these sensors that you can see there. So they provide [INAUDIBLE] and also [INAUDIBLE]. And they're mounted in each of the limbs of the robot and the torso. And they allow the robot to [INAUDIBLE] interaction with the world.

And indeed, with this kind of object, it can do many different things. For instance in this video, I'm showing an example of how the feedback provided by the force/torque sensor can be used to guide the robot and teach it to learn different kind of action. Like, in this case, a pouring action and then repeat it and maybe try to generalize it.

So force/torque sensors provide the robot feedback about the intensity of the interaction with the external world. But they're not allowed to have the robot have an idea of where this kind of interaction is occurring. So, for this, we have artificial skin covering the robot, as I told you.

And the technology used for the kind of thing, that you can see here, for the palm of the hand of the robot, is capacitive and it's similar to the technology used for smart phones. It's using, if you can see these yellow dots. They are all electrodes that, together with another arm, they form a capacitor. The way the arm works of this capacitor are formed when there is an interaction with the environment allowed to provide the feedback of the kind of intensity [INAUDIBLE] the location itself.

It's providing information about where this interaction is occurring. And the artifical skin is actually really useful for an embodied agent. And for reasons like we see in this video, without using this feedback, if you have a very light object, the robot is not able to detect the process is interrupting with something. It's just closing the end and it doesn't have any feedback. It's crushing the object. By using the sensors on the fingertips of the hand, the robot is able to detect that it's actually touching something, and therefore, it's stopping the action without crushing the object.

So other useful things that can be done with artificial skin. This is an example of combining the information from the force/torque sensor and the artificial skin. So the artificial skin allows to detect where and the direction of the kind of force that is applied to the robot. And in that case, the robot is counterbalancing. It's negating the effect of gravity and of internal forces.

So it's basically having arm floating around like if it was in space. And there is no friction. So by touching the arm of the robot, we have the arm drifting in the direction opposite to where the force is applied, or the torque. You can see that, the arm is turning. But as you can see, like it was in space without any friction, because it's actually negating both gravity and internal forces.

And finally, again about the artificial skin, there's some work by Alessandro, but he's going to talk about something else, but I found is particularly interesting to show him. This is an example of the robot self-calibrating the model of its own body, with respect to its own hand. The idea is to have the robot use the tactile feedback from the fingertip and the skin of its forearm, for instance, to learn the position between the fingertip and the arm.

And therefore, it's able, first by touching itself, to learn the correspondence and then to actually show that it has learned this kind of correlation by reaching the point when someone else touches the thing point. And tries to reach it. And therefore, this can be seen as a way of self-calibrating without the need of a model of the kinematics of the robot. The robot would just be able to explore itself and learn how different part of it's body relate one to the other.

And again, related to self-calibration sometimes, but this is more of a calibration between vision and motor activity. This is a work appeared in 2014 in which, basically, the correlation between the kind of actions that the robot is able to perform is calibrated with respect to its ability to perceive the work. So, in this kind of video I'm going to show, the robot is trying to reach for an object and failing in its action because the actual model of the word that you use to perform the reaching is not aligned with the 3-D model provided by vision. And this can happen due to smaller errors in the kinematics or in the vision and therefore even just small errors cause a complete failure of the system.

And therefore, by trying to correlate that information, the one from the kinematics, in this case the robot is using it on the fingertip to the see where it actually is in the image. And when the kinematic model predicts the hand should be. So the green dot is the point, predicted by kinematics, of where the system expects the fingertip to be, and where actually the fingertip is. Of course, by learning this relation, the robot is able to cope with this kind of misalignment. And therefore, after a first calibration phase, it's able to perform the reaching action, successfully, as you will see in a moment.

Also, this kind of ability of calibrating would be pretty useful in case of the situations in which the robot is damaged, and therefore, it's actual model changes completely. As you can see, now it's reaching and it's performing the grasp correctly. Finally, before going with the actual talk, I'm going to show a final video about balancing. Some of you have asked if the robot works, and the robot is currently not able to work, but this is a video from the people from the group that is actually in charge of making the robot work. The first step, of course, will be balancing. And this is an example of it.

It's actually one foot balancing where multiple components of what I've shown you about the iCub so far. Torque sensing, initial sensing are combined together to have the robot stand on one foot. And also be able to cope with the interaction with internal forces that could try to have it fall. They are applying forces to the robot and it's able to detect the force and to cope a bit with the forces but to stay stable. OK so, this was just a brief overview of some things that can be done with the iCub. And actually, the next talk will be a bit more about what's going on with it.

ALESSANDRO RONCONE: I want to talk to you about part of my PhD project that was about tackling the perception problem. Tackling the perception problem through the use of multisensor integration. And specifically, I narrowed down this big problem by implementing a model of PeriPersonal Space on the iCub. That is, biology inspired approach. PeriPersonal Space is a concept that has been known in neuroscience and psychology for years.

And so, let me start with what PeriPersonal Space is, and why it is so important for humans and animals. It is defined as the space around us, within which objects can be grasped and manipulated. It's an interface, basically, between our body and the external world. And for this reason, it benefits from a multimodal integrated representation that merges information between different modalities. And historically, these have been the vision system, the tactile system, the perception, auditory system, and even the motor system.

Historically, it has been studied by two different fields. The neurophysiology on one side, and all that could be related to psychology and developmental psychology on the other. They basically follow the two different approaches, the former being bottom up, the latter being top down. And they came out with different outcomes. And the former emphasizes the role of the perception and it interplays with the motor system in the control of movement, whereas the latter was focusing mainly on the multisensory aspect, that is, how different modalities were combined together in order to form a coherent view of the body and the nearby space.

Luckily, in recent years, they decided to converge to a common ground, and a shared interpretation, and for the purposes of my work I would like to highlight the main aspect. Firstly, and this one might be of interest from an engineering perspective. PeriPersonal Space is made of different reference points that are located in different regions of the brain. And there might be a way for the brain to switch from one to another, according to different contact and goal. And secondly, as I was saying, PeriPersonal Space benefits from a multisensory integration in order to form a coherent view of the body understanding space.

In this experiment made by Fogassi in 1996, they basically found a number of so-called visual tactile neurons that are set up neurons that fire both stimulated in a specific skin part and if an object is presented in surrounding space. So this means that these neurons code both the visual information and the tactile information. But they also have some proprioceptive information, because they are basically attached to the body part that they belong to. Lastly, one of the main properties of this presentation is in basic plasticity.

And for example, in this experiment, made by Iriki ten years ago, the extension of this receptive field in the visual space, in the surrounding space, after training with a rake, have been shown to go up to enclose the tool as if this tool becomes part of the body. So through experience and through tool use, the monkey was able to grow this receptive field.

Those are properties that are very nice, and we would like them to be available for the robot. And, in general robotics, the work related to PeriPersonal Space can be divided into two groups.

On the one side, the model, and the simulation basically. The closest one to my work was the one from Fuke, a colleague that are from [INAUDIBLE] lab, in which they used a stimulated robot in order to model the mechanisms that are leading to this visual-tactile presentation. On the other side, there are the engineering approaches that are few. The closest one is this one by Mittendorfer from Gordon Cheng's lab, in which they first developed the multimodal skin. So they developed the hardware to be able to do that.

And then they use the to trigger local avoidance responses, reflexes to incoming objects. We are trying to position ourself in the middle. Let's say, we are not trying to create a perfect model of PeriPersonal Space from a biological perspective. But on the other side, we would like to have something that is also working, and useful for our proposals. So from now on, I will divide the presentation in two parts. The first will be about the model.

So what we think will be useful for tackling the problem, on the other side I show you an application of this model, that is basically using the low-color presentation in order to trigger avoidance responses or reaching responses distributed throughout the body. So let me start with the proposed model of PeriPersonal Space. Loosely inspired by the neurophysiological findings like we discussed before, we developed this PeriPersonal Space presentation by means of access of facial receptive field that we are going out from the robot's skin. So basically, they were extending the tactile domain into nearby space. Each tactile, that is, each pair the iCub skin is composed of, will experience a set of multisensory events.

So basically, you are letting the robot learn this visual-tactile sensations by taking an object and making contact on the skin part. We learn it by tactile experience we learn a sort of probability of being touched prior to contact activation when the new incoming object is presented. And we basically created this cone shape receptive field that is going from each of the taxels. And for any object that is entering this receptive field, we created we called a buffer, of the path so basically, the idea is that the orbit has some information from what was going on before they touch, the actual contact.

And if the object eventually ends up touching the tactile, it will be labeled as a positive event that will enforce the probability of the event the ending of touching the taxel. If not, for example, it might be that the object enters this receptive field, and in the end, ends up touching another taxel. This will be labeled as a negative. So at the end, we will have a set of positive and negative events a taxel can learn from. This is three dimensional space because the distance is three dimensional.

And we narrowed it down to a one dimensional domain, by basically, taking the norm of the distance, but also the relative position of the object and the taxel. In order for us to be able to cope with the calibration errors that were amounting up to a couple of centimeters that were significant. One dimensional variable has been discretized into a set of bins. And for each bin, we computed the probability of an event belonging to that, of being touched. So the idea is that, at 20 centimeters, the probability of being touched would be lower than a zero centimeter. This is the intuitive idea.

Over this one dimensional visualization, we used a partial window interpolation technique in order to provide us with the two dimensional function that, at the end, we give up a inactivation value that is proportional with the distance of the object. So as soon as the new object will enter the receptive field, I will have the taxel fire before being contacted. We did, basically, two experiments. Initially, we did a simulation in a mock lab in order to assess the convergence on the long term learning, one-shot learning behavior, to assess if our model was able to cope with noise, with the current calibration errors.

And then, we went on the real robot. We presented them with different objects. And we were basically touching the robot 100 times in order to make it learn these presentations. So, trust me, I don't want to bother you with this kind of technicalities, but we did a lot of work. This is, basically, the math of the result.

So, let me go on the second part, in which the main problem was for the robot to detect the object visually. In order for us to do that, we developed a 3D tracking algorithm, that was able to track a [INAUDIBLE] object, basically.

To design, we used some software that was only available in the iCub software repository. The engine provides you with some basic algorithms that you can play with. And namely, we used a two dimensional optical flow made by Carlo and a 2D particle filter and a 3D stereo vision algorithm, that is basically the same as I was showing before during the recognition game.

And this basically was feeding a 3-D camera to provide the robot estimation of the position of the object. So, the idea is that the motion detector from the optical flow act as a trigger for the subsequent pipeline, in which, basically, after a consistent enough motion in this optical flow module, this would be a template to be taught in the visual in the 2D visual world by this. Then, that this information is sent to the 3D depth map and this would be feeding the camera feature in order to provide us with the table representation because, obviously, the stereo system doesn't work that good in our context. And this, if it works-- no. OK. On my laptop, it works here. OK. Now it works. OK. This is the idea.

So I was basically waving, moving the object in the beginning. OK. Then when it is detected, this pattern starts. And you can see here the tracking. This is the stereo vision. This the final outcome.

This was used for the learning. We did a lot of iterations of these objects that are approaching the skin on different body parts. This is the graph. I don't want to talk about that. So let me start with the video.

This is basically the skin. And this is the part that it was trained before. When there is a contact, there is activation here. You can see here the activation.

And soon after, this thing worked also with one example. The taxel starts firing before the contact. And obviously, this is improved over the time.

And it depends on the body part that is touched. For example, if I touch here, I'm coming from the top. So the representation starts firing mainly here.

And this, obviously, depends on the specific body part. Now, I think that I'm going to touch the hand. And so after a while, you will have an activation on the hand. Obviously, I will have also some activation in the forearm, because I was getting closer to the forearm.

And as an application of this, this one is simply a presentation. So it's not that usable. We basically exploited it in order to develop an avoidance-- a margin of safety around the body. Let's say if the taxel is firing, I would like it to go away from the object, assuming that this can be a potentially harmful object.

And on the other way, I would like it to be able to reach with any body part the object under consideration. So to this end, we developed the avoidance and catching controller that was able to leverage on this distributed information and perform a sensor-based guidance of the model actions by means of this visual tactile associations. And this is basically how it works.

So this is at the testing stage. So I already learned the representation. As soon as I get closer, the taxel starts firing, because of the probabilities I was learning. And the arm goes away.

Obviously, the movement depends on the specific skin part that has been touched. If I'm touching here, the object will go away from here. If I'm coming from the top-- I think this one was doing from the top, yes-- the object will go away from the back. The object will be going a way from another direction.

And the idea here is not to, basically, tackle the problem from a classical robotics approach. But the basic idea-- this behavior emerges from the learning. And the idea was very simple.

We were basically looking at the taxel that we were firing. If they were firing enough, then we were recording their position. And we were doing, basically, a population coding that is a weighted average according to the activation and the prediction.

We did that to both for the position of the taxel and the normal. So at the end if you have a bunch of tactiles here, we will end up with one point to go away from. And on the other side, the catching, the reaching, was basically the same, but in the opposite direction.

So if I want to avoid, I do this. If I want to catch, I do this. Obviously, if you do it in the hand, this would be a standard robotic reaching. But this actually can be triggered also in different body parts. As you can see here, I get a virtual activation, and then the physical contact. And yes, basically, our design was to use the same controller for both of the behaviors.

OK. This is also some technicalities that I don't want to show you. So in conclusion, the detector presented here is, to our knowledge, the first attempt at creating a decentralized, multisensory visual tactile representation for a robot and its nearby space by means of the distributed skin and interaction with the environment. One of the assets of our representation is that learning is fast.

As you were seeing, it can learn, also, from one single example. It's in parallel for the whole body in the sense that every tactile learns independently. Its own representation is incremental in a sense that it converges toward a stable representation over the time.

And importantly, it is adapted from experience. So basically, it can automatically compensate for errors in the model that, for humanoid robots, is one of the main problems when merging different modalities OK. Thank you. If you have any question, feel free to ask.

RAFFAELLO CAMORIANO: I am Raffaello. And today, I'll talk to you about a little bit of my work on machine learning and robotics, in particular some subsets of machine learning which are the large scale learning and incremental learning. But what do we expect from our modern robot? And how can machine learning help out with this?

Well, we expect modern robots to work in, particularly, unstructured environments which they have never seen before and to learn new tasks on the fly depending on the particular needs throughout the operation of the robot itself and across different modalities. For instance-- vision, of course, but also tactile sensing which is available on the iCub also proprioceptive sensing, including force sensing, [INAUDIBLE] and so on and so forth. And we want to do all of this throughout a very long time span potentially. Because we expect robots to be companions of humans in the real world operating for maybe years or more.

And this poses a lot of challenges, especially from the computational point of view. And machine learning can actually help with this tackling these challenges. For instance, there are large scale learning methods, which are algorithms which can work with very large scale datasets.

For instance, if we have millions of points gathered by the robot cameras throughout 10 days and we want to process them, well, if we use standard machine learning methods, that will be a very difficult problem to solve if we don't use, for instance, randomizing methods and so on and so forth. Machine learning also has incremental algorithms, which can allow the learned model to be updated as new previously unseen features are presented to the agent. And also, there is a subfield of transfer learning which allows knowledge learned for a particular task to be used for serving another related task without the need for seeing many new examples for the new task.

So my main research focuses are in machine learning. I work especially in large scale learning methods, incremental learning, and in the design of algorithms which allow for computational and accuracy trade-offs. I will explain this a bit more later.

And as concerns robotic applications, I work with Guilia, Carlo and others on incremental object recognition, so in a setting in which the robot is presented new objects throughout a long time span. And it has to learn them on the fly. And also, I'm working in a system identification setting, which I will explain later, related to the motion of the robot.

So this is one of the works which has occupied my last year. And it is related to large scale learning. So if we consider that we may have a very large n, which is a number of examples we have access to, in the setting of kernel methods, we may have to store a huge matrix, the matrix K, which is n by n, which could be simply impossible to store.

So there are randomized methods, like the Nystrom method, which enable to compute a low rank approximation of the kernel metrics simply by throwing a few points m at random, a few samples at random, and building the metrics K and m, which is just much smaller. Because m is much smaller than n. And this is a well-known method in machine learning.

But we tried to see it from a different point of view than usual. Usually, this is seen just from a computational point of view in order to fit a difficult problem inside computers with limited capabilities while we proposed to see the Nystrom approximation as regularization of operation itself. So if you can see this, the usual way in which the Nystrom method is applied, for instance, with kernel regularized least squares.

The parameter m, so the number of examples we are taking at random, is usually taken as large as possible in order just to fit in the memory of the available machines. While, actually, after choosing a large m, it is often necessary to regularize with deep neural regularization, for instance. And this sounds a bit like a waste of time and memory.

Because, actually, what regularization, roughly speaking, does is to discard the irrelevant eigen components of the kernel metrics. So we observe that we can do this by just less random examples, so having a smaller model which can be computed more efficiently and without having to regularize again later. So m, the number of examples which are used, controls both the regularization and the computational complexity of our algorithm.

This is very useful in a robotic setting in which we have to deal with lots of data. As regards to the incremental objects recognition task, this is another project I'm working on. And imagine that the robot has to work in an unknown environment, and it is presented novel objects on the fly. And it has to update its object recognition model in an efficient way without retraining from scratch every time a new object arrives.

So this can be done easily by a slight modification of the regularized least squares algorithm and proper reweighting. An open question is how to change the regularization as n grows. Because we didn't find yet a way to efficiently update regularization parameter in this case. So we are still working on this.

The last project I'll talk about is more related to, let's see, physics and motion. So we have an arbitrary limb of the robot, for instance, the arm. And our task is to learn a model which can provide an interesting dynamics model. So it can predict the inner forces of the arm during motion.

This is useful, for instance, in a contact detection setting. So when the sensor readings are different from the group predicted one, that means that there may be a contact. Or for external force estimation or, for example, for the identification of the mass of a manipulated object.

So we have some challenged for this project. We have to devise a model which could be interpretable, so in which the rigid body dynamic parameter would be understandable and intelligible for controlled purposes. And we wanted this model to be more accurate than standard multibody dynamics, rigid body dynamics model.

And also, we want to adapt to changing conditions throughout time. For instance, during the operation of the robot, after one hour, the changes in temperature determine a change also in the dynamic properties of the mechanical properties of the arm. And we want to accommodate for this in an incremental way.

So this is what we did. We implemented a semi-parametric model which the first part which has priority is a simple incremental parametric model. And then we used random features for building non-parametric incremental model which can be updated in an efficient way.

And we shown with this real experiment that the semi-parametric model worked as well as the non-parametric one. But it's faster to converge, because it has an initial knowledge about the physics of the arm. And it is also better than the fully parametric one, because it also models, for example, dynamical effect due to deflectability of the body. And dynamic deflectors are usually not modeled by rigid body dynamic models.

OK. Another thing I'm doing is maintaining the Grand Unified Regularized Least Squares library, which is a library for regularized least squares, of course. It supports a large scale dataset. This was developed in joint exchange between MIT and IIT some years ago by others, not by me. And it has a MATLAB and a C++ interface. If you want to have a look at how these methods work, I suggest you to try out tutorials which are available on GitHub.

GUILIA PASQUALE: I'm Guilia. And I work on the iCub robot with my colleagues, especially on vision and, in particular, on visual recognition. I work under the supervision of Lorenzo Natale and Lorenzo Rosasco. Both will be here for a few days in the following weeks.

And the work that I'm going to present has been done in collaboration with Carlo and also Francesca Odone from the University of Genoa. So in the last couple of years, computer vision methods based on deep convolution on neural networks have achieved a remarkable performance in tasks such as large scale image classification and retrieval. And the extreme success of these methods is mainly due to the increasing availability of all these larger datasets.

And in particular, I'm referring to the ImageNet one, which is composed by millions of examples labeled into thousands of categories through crowd sourcing methods such as the Amazon Turk. And in particular, the increased data availability to gather with the increased computational power has allowed to train deep networks characterized by millions of parameters in a supervised way from the image up to the final label through the back propagation algorithm.

And this has allowed to mark a breakthrough-- in particular, in 2012 when Alex Krizhevsky proposed for the first time of a network of this kind trained on ImageNet dataset and definitely won the ImageNet large scale user recognition challenge in this way. And the trend has been confirmed in the following years. So that nowadays problems such as large scale image classification or detection are usually tackled following this deep learning approach.

And not only, it has been also demonstrated at least empirically. Oh, I'm sorry. Maybe this is not particularly clear. But this is the Krizhevsky Network.

Models of networks of this kind trained on large datasets, such as the ImageNet one, do provide also very good general and powerful image descriptors to be applied also on other tasks and datasets. In particular, it is possible to use a convolutional neural network trained on ImageNet dataset, feed it with images, and using it as a black box extracting the vectorial representation of the incoming images as the output of one of the intermediate layers. Or even better, it is possible to start from a network model trained on the ImageNet dataset and fine tune its parameters on a new dataset for a new task and achieving and surpassing the state of the art-- for example, also in the Pascal dataset and other tasks-- following this approach.

So it is natural to ask at this point, why? Instead, in robotics, providing robots with robust and accurate visual recognition capabilities in the real world is still one of the greatest challenge that prevents the use of autonomous agents for concrete applications. An actually, this is a problem that is not only related to the iCub platform, but it is also a limiting factor that the performance of the latest robotics platforms, such as the ones that have been participating, for example, to the DARPA robotics challenge.

Indeed, as you can see here, robots are still either highly tele-operated or complex methods.

To, for example, map the 3D structure on the environment and label it a priori must be implemented in order to enable autonomous agents to act in very controlled environments. So we decided to focus on very simple settings where, in principle, computer vision methods as the ones that I've been describing you should be at least-- well, should provide very good performances. Because here the setting is pretty simple.

And we tried to evaluate the performance of these deep learning methods in these settings. Here you can see the robot, that one, standing in front of a table. There is a human which gives verbal instruction to the robot and also, for example in this case, the label of the object to be either learned or recognized.

And the robot can focus his attention on potential objects through bottom up segmentation techniques-- for example, in this case, color or the other saliency-based segmentation methods. I'm not going into the detail of this setting, because you would see a demo after my talk of this. Another setting that we are considering is similar to the previous one.

But this time, there is a human standing in front of the robot. And there is no table. And the human, he's holding the objects in his hands and is showing one object after the other to the robot providing the verbal annotation for that object.

The robot in this way, for example here, can exploit motion detection techniques in order to localize the object in the visual field and focus on it. The robot tracks the object continuously, acquiring in this way cropped the frames around the object that are the training examples that will be used to learn the object's appearance. So in general, this is the recognition pipeline that is implemented to perform both the two behaviors that I've been showing you.

As you can see, the input is the image, the stream of images from one of the two cameras. Then there is the verbal supervision of the teacher. Then there are segmentation techniques in order to crop region of interest from the incoming frame and feed this crop to a convolutional neural network.

In this case, we are using the famous Krizhevsky model. Then we encode each incoming crop in a vector as the output of one of the latest layers of the network. And we feed all these vectors to a linear classifier, which is linear because, in principle, the representation that we are extracting is good enough for the discrimination that we want to perform.

And so the classifier uses these incoming vectors either as examples for the training sector or assigns to each vector the prediction label. And the output is an histogram with the probabilities of all the classes. And the final outcome is the one with the highest probability. And the histogram is updated in real time.

So this pipeline can be used either for one or the other settings that have been described you. So in particular, we started from trying to list some requirements that according to us are fundamental in order to implement a sort of ideal robotic visual recognition system. And these requirements are usually not considered by typical computer vision methods as the ones that have been described you, but are the same fundamental if we want to achieve human level performances in the settings that I've been showing you.

For example, first of all, the system should be, as you have seen, as much as possible self-supervised, meaning that there must be techniques in order to focus that robot's attention on the object of interest and isolate them from the visual field. Then hopefully, we would like to come out with a system that is reliable and robust to the variations in the environment and also in the object's appearance. Then also, as we are in the real world, we would like a system able to exploit the contextual information that is available.

For example-- the fact that we are actually dealing with videos. So the frames are temporarily correlated. And we are not dealing with images in the wild, as the ImageNet case.

And finally, as Raffaello was mentioning, we would like to have a system that is able to learn incrementally to build always richer models of the object through time. So we decided to evaluate this recognition pipeline according to the criteria that have been described you. And in order to provide reproducibility to our study, we decided to acquire a dataset on which to perform our analysis.

However, we would like also to be confident enough that the result that we obtain on our benchmark will hold also in the real usage of our system. And this is the reason why we decided to acquire our dataset in the same application setting where the robot usually operates. So this is the iCubWork28 dataset that I acquired last year.

As you can see, it's composed by 28 objects divided into seven categories and four instances per category. And I acquired it for four different days in order to test also incremental learning capabilities. The dataset is available on the IIT website. And you can also use it, for example, for the project of trust five if you are interested.

And this is an example of the kind of videos that I acquired considering one of the 28 objects. There are four videos for the train, four for the test, acquired in four different conditions. The object is undergoing random transformations, mainly limited to 3D rotations.

And as you can see, the difference between the days is mainly limited to the fact that we are just changing the conditions in the environment-- for example, the background or the lighting conditions. And we acquired eight videos for each of the 28 objects that I show you. So first of all, we tried to find a measure, as I was saying before, to quantify the confidence with which we can expect that the results and the performance that we observe on these benchmarks will also hold in the real usage of the system.

And to do this, first of all, we focused only on object identification for the moment. So the task is to discriminate the specific instances of objects among the pool of 28. And we decided to estimate for an increasing number of objects to be discriminated from 2 to 28 the empirical probability distribution of the identification accuracy that we can observe statistically for a fixed number of objects. That is depicted here in the form of box plots.

And also, we estimated for each fixed number of objects to be discriminated the minimum accuracy that we can expect to achieve with increasing confidence levels. And this is a sort of data sheet. The idea is to provide an idea to an hypothetical user of the robot of the identification accuracy that can be expected given a certain pool of objects to be discriminated.

So the second point that I'll briefly describe you is the fact that we investigated the effect of having a more or less precise segmentation in the image. So we evaluated the task of identifying the 28 objects with different levels of segmentation starting from the whole image up to a very precise amount of segmentation of the objects. It can be seen that, indeed, even if in principle these convolutional networks are trained to classify objects in the world image as it is in the ImageNet dataset, it is still true that in our case we observed that there is still a large benefit from having a fine-grained segmentation.

So probably the network is not able to completely discard the new relevant information that is in the background. So this is a possible interesting direction of research. And finally, the last point that I decided to tell you-- I will skip on the incremental part, because it's an ongoing work that I'm doing with Raffaello-- is about the exploitation of the temporal contextual information.

Here, you can see the same kind of plot that I showed you before. So the task is object identification, increasing number of objects. And the dot black line represent the accuracy that you obtain if you consider, as you were asking before, the classification of each frame independently.

So you can see that in this case the accuracy that you get is pretty low, considering that we have to discriminate between only 28 objects. However, it is also true that as soon as you start considering instead of the prediction given looking only at the current frame, the most frequent prediction occurred in a temporal window, so in the previous, let's say, 50 frames. You can boost your condition accuracy a lot.

As you can see here, from green to red, increasing the length of the temporal window increases the recognition accuracy that you get. This is a very simple approach. But it is showing that actually it is relevant in the fact that you are actually dealing with videos instead of images in the wild. And it is another direction of research.

So finally, in the last part of my talk, I would like to tell you about the work that I'm actually doing now, which is concerning about most object categorization tasks instead of identification. And this is the reason why we decided to acquire a new dataset, which is larger than the previous ones. Because it is composed not only by more categories, but, in particular, by more instances per category in order to be able to perform categorization experiments as I told you.

Here, you can see the categories with which we are starting are 28 divided into seven macro categories, let's say. But the idea of this dataset is to have a continuously expandable in time dataset. So there is an application that we used to acquire these datasets. And the idea is to perform periodical acquisitions in order to incrementally enrich the knowledge of the robot about the objects in the scene.

Also, another important factor regarding this dataset is that differently from the previous one this would be divided and tagged by nuisance factors. And in particular, for each object, we are acquiring different videos where we isolate the different transformations that the object is undergoing. So we have a video where the object is just at different scales.

Then it is rotating on the plane. Outside the plane, it is translating. And then there is a final video where all of this transformation occurs simultaneously.

And finally, to acquire this dataset we decided to use the depth information, so that in the end we acquired both the left and the right to cameras. And in principle, this information could be used to obtain the 3D structure of the objects. And this is the idea that we used in order to make the robot focusing on the object of interest using disparity.

Disparity is very useful in this case, because it allows to detect unknown objects just given the fact that we know that we want the robot focused on the closest objects in the scene. So it is a very powerful method in order to have the robot tracking a known object with all different lighting conditions and so on. Yeah. And here, you can see this is the left camera.

This is the disparity map. This is its segmentation, which provides an approximate region of interest around the object. And this is the final output.

So I started acquiring the first-- well, it should be red, but it's not very clear I mean. I started acquiring the first categories among these 21 listed here, which are the squeezer, sprayer, the cream, the oven glove, and the bottle. For each row, you see the tiny instances that I collected.

And the idea is to continue acquiring them when I come back in Genoa. And so here, you can see an example of the five videos. Actually, I acquired 10 videos per object, five for the training set and five for the test set.

And you can see that in the different videos the object is undergoing different transformations. And this is the final one while these transformation are mixed. Oh, the images here are not segmented yet. So you can see the whole image. But in end, also the information about the segmentation and disparity and so on will be available.

And this dataset regarding the 50 object that I acquired together with the application that I'm using to acquire the dataset are available if you are willing to use them for the projects in trust five, for example, in order to investigate the invariant properties of the different representations. And so that's it.