Lecture 1 - Introduction to Convolutional Neural Networks for Visual Recognition

So welcome everyone to CS231n.

I'm super excited to offer this class again for the third time.

It seems that every time we offer this class it's growing exponentially unlike most things in the world.

This is the third time we're teaching this class.

The first time we had 150 students.

Last year, we had 350 students, so it doubled.

This year we've doubled again to about 730 students when I checked this morning.

So anyone who was not able to fit into the lecture hall I apologize.

But, the videos will be up on the SCPD website within about two hours.

So if you weren't able to come today, then you can still check it out within a couple hours.

So this class CS231n is really about computer vision.

And, what is computer vision?

Computer vision is really the study of visual data.

Since there's so many people enrolled in this class, I think I probably don't need to convince you that this is an important problem, but I'm still going to try to do that anyway.

The amount of visual data in our world has really exploded to a ridiculous degree in the last couple of years.

And, this is largely a result of the large number of sensors in the world.

Probably most of us in this room are carrying around smartphones, and each smartphone has one, two, or maybe even three cameras on it.

So I think on average there's even more cameras in the world than there are people.

And, as a result of all of these sensors, there's just a crazy large, massive amount of visual data being produced out there in the world each day.

So one statistic that I really like to kind of put this in perspective is a 2015 study from CISCO that estimated that by 2017 which is where we are now that roughly 80% of all traffic on the internet would be video.

This is not even counting all the images and other types of visual data on the web.

But, just from a pure number of bits perspective, the majority of bits flying around the internet are actually visual data.

So it's really critical that we develop algorithms that can utilize and understand this data.

However, there's a problem with visual data, and that's that it's really hard to understand.

Sometimes we call visual data the dark matter of the internet in analogy with dark matter in physics.

So for those of you who have heard of this in physics before, dark matter accounts for some astonishingly large fraction of the mass in the universe, and we know about it due to the existence of gravitational pulls on various celestial bodies and what not, but we can't directly observe it.

And, visual data on the internet is much the same where it comprises the majority of bits flying around the internet, but it's very difficult for algorithms to actually go in and understand and see what exactly is comprising all the visual data on the web.

Another statistic that I like is that of Youtube.

So roughly every second of clock time that happens in the world, there's something like five hours of video being uploaded to Youtube.

So if we just sit here and count, one, two, three, now there's 15 more hours of video on Youtube.

Google has a lot of employees, but there's no way that they could ever have an employee sit down and watch and understand and annotate every video.

So if they want to catalog and serve you relevant videos and maybe monetize by putting ads on those videos, it's really crucial that we develop technologies that can dive in and automatically understand the content of visual data.

So this field of computer vision is truly an interdisciplinary field, and it touches on many different areas of science and engineering and technology.

So obviously, computer vision's the center of the universe, but sort of as a constellation of fields around computer vision, we touch on areas like physics because we need to understand optics and image formation and how images are actually physically formed.

We need to understand biology and psychology to understand how animal brains physically see and process visual information.

We of course draw a lot on computer science, mathematics, and engineering as we actually strive to build computer systems that implement our computer vision algorithms.

So a little bit more about where I'm coming from and about where the teaching staff of this course is coming from.

Me and my co-instructor Serena are both PHD students in the Stanford Vision Lab which is headed by professor Fei-Fei Li, and our lab really focuses on machine learning and the computer science side of things.

I work a little bit more on language and vision.

I've done some projects in that.

And, other folks in our group have worked a little bit on the neuroscience and cognitive science side of things.

So as a bit of introduction, you might be curious about how this course relates to other courses at Stanford.

So we kind of assume a basic introductory understanding of computer vision.

So if you're kind of an undergrad, and you've never seen computer vision before, maybe you should've taken CS131 which was offered earlier this year by Fei-Fei and Juan Carlos Niebles.

There was a course taught last quarter by Professor Chris Manning and Richard Socher about the intersection of deep learning and natural language processing.

And, I imagine a number of you may have taken that course last quarter.

There'll be some overlap between this course and that, but we're really focusing on the computer vision side of thing, and really focusing all of our motivation in computer vision.

Also concurrently taught this quarter is CS231a taught by Professor Silvio Savarese.

And, CS231a really focuses is a more all encompassing computer vision course.

It's focusing on things like 3D reconstruction, on matching and robotic vision, and it's a bit more all encompassing with regards to vision than our course.

And, this course, CS231n, really focuses on a particular class of algorithms revolving around neural networks and especially convolutional neural networks and their applications to various visual recognition tasks.

Of course, there's also a number of seminar courses that are taught, and you'll have to check the syllabus and course schedule for more details on those 'cause they vary a bit each year.

So this lecture is normally given by Professor Fei-Fei Li.

Unfortunately, she wasn't able to be here today, so instead for the majority of the lecture we're going to tag team a little bit.

She actually recorded a bit of pre-recorded audio describing to you the history of computer vision because this class is a computer vision course, and it's very critical and important that you understand the history and the context of all the existing work that led us to these developments of convolutional neural networks as we know them today.

I'll let virtual Fei-Fei take over [laughing] and give you a brief introduction to the history of computer vision.

Okay let's start with today's agenda.

So we have two topics to cover one is a brief history of computer vision and the other one is the overview of our course CS 231 so we'll start with a very brief history of where vision comes from when did computer vision start and where we are today.

The history the history of vision can go back many many years ago in fact about 543 million years ago.

What was life like during that time?

Well the earth was mostly water there were a few species of animals floating around in the ocean and life was very chill.

Animals didn't move around much there they don't have eyes or anything when food swims by they grab them if the food didn't swim by they just float around but something really remarkable happened around 540 million years ago.

From fossil studies zoologists found out within a very short period of time — ten million years — the number of animal species just exploded.

It went from a few of them to hundreds of thousands and that was strange — what caused this?

There were many theories but for many years it was a mystery evolutionary biologists call this evolution's Big Bang.

A few years ago an Australian zoologist called Andrew Parker proposed one of the most convincing theory from the studies of fossils he discovered around 540 million years ago the first animals developed eyes and the onset of vision started this explosive speciation phase.

Animals can suddenly see; once you can see life becomes much more proactive.

Some predators went after prey and prey have to escape from predators so the evolution or onset of vision started a evolutionary arms race and animals had to evolve quickly in order to survive as a species so that was the beginning of vision in animals after 540 million years vision has developed into the biggest sensory system of almost all animals especially intelligent animals in humans we have almost 50% of the neurons in our cortex involved in visual processing it is the biggest sensory system that enables us to survive, work, move around, manipulate things, communicate, entertain, and many things.

The vision is really important for animals and especially intelligent animals.

So that was a quick story of biological vision.

What about humans, the history of humans making mechanical vision or cameras?

Well one of the early cameras that we know today is from the 1600s, the Renaissance period of time, camera obscura and this is a camera based on pinhole camera theories.

It's very similar to, it's very similar to the to the early eyes that animals developed with a hole that collects lights and then a plane in the back of the camera that collects the information and project the imagery.

So as cameras evolved, today we have cameras everywhere this is one of the most popular sensors people use from smartphones to to other sensors.

In the mean time biologists started studying the mechanism of vision.

One of the most influential work in both human vision where animal vision as well as that inspired computer vision is the work done by Hubel and Wiesel in the 50s and 60s using electrophysiology.

What they were asking, the question is "what was the visual processing mechanism like in primates, in mammals" so they chose to study cat brain which is more or less similar to human brain from a visual processing point of view.

What they did is to stick some electrodes in the back of the cat brain which is where the primary visual cortex area is and then look at what stimuli makes the neurons in the in the back in the primary visual cortex of cat brain respond excitedly what they learned is that there are many types of cells in the, in the primary visual cortex part of the the cat brain but one of the most important cell is the simple cells they respond to oriented edges when they move in certain directions.

Of course there are also more complex cells but by and large what they discovered is visual processing starts with simple structure of the visual world, oriented edges and as information moves along the visual processing pathway the brain builds up the complexity of the visual information until it can recognize the complex visual world.

So the history of computer vision also starts around early 60s.

Block World is a set of work published by Larry Roberts which is widely known as one of the first, probably the first PhD thesis of computer vision where the visual world was simplified into simple geometric shapes and the goal is to be able to recognize them and reconstruct what these shapes are.

In 1966 there was a now famous MIT summer project called "The Summer Vision Project.

" The goal of this Summer Vision Project, I read: "is an attempt to use our summer workers effectively in a construction of a significant part of a visual system.

" So the goal is in one summer we're gonna work out the bulk of the visual system.

That was an ambitious goal.

Fifty years have passed; the field of computer vision has blossomed from one summer project into a field of thousands of researchers worldwide still working on some of the most fundamental problems of vision.

We still have not yet solved vision but it has grown into one of the most important and fastest growing areas of artificial intelligence.

Another person that we should pay tribute to is David Marr.

David Marr was a MIT vision scientist and he has written an influential book in the late 70s about what he thinks vision is and how we should go about computer vision and developing algorithms that can enable computers to recognize the visual world.

The thought process in his, in David Mars book is that in order to take an image and arrive at a final holistic full 3d representation of the visual world we have to go through several process.

The first process is what he calls "primal sketch;" this is where mostly the edges, the bars, the ends, the virtual lines, the curves, the boundaries, are represented and this is very much inspired by what neuroscientists have seen: Hubel and Wiesel told us the early stage of visual processing has a lot to do with simple structures like edges.

Then the next step after the edges and the curves is what David Marr calls "two-and-a-half d sketch;" this is where we start to piece together the surfaces, the depth information, the layers, or the discontinuities of the visual scene, and then eventually we put everything together and have a 3d model hierarchically organized in terms of surface and volumetric primitives and so on.

So that was a very idealized thought process of what vision is and this way of thinking actually has dominated computer vision for several decades and is also a very intuitive way for students to enter the field of vision and think about how we can deconstruct the visual information.

Another very important seminal group of work happened in the 70s where people began to ask the question "how can we move beyond the simple block world and start recognizing or representing real world objects?

" Think about the 70s, it's the time that there's very little data available; computers are extremely slow, PCs are not even around, but computer scientists are starting to think about how we can recognize and represent objects.

So in Palo Alto both at Stanford as well as SRI, two groups of scientists that propose similar ideas: one is called "generalized cylinder," one is called "pictorial structure.

" The basic idea is that every object is composed of simple geometric primitives; for example a person can be pieced together by generalized cylindrical shapes or a person can be pieced together by critical part in their elastic distance between these parts so either representation is a way to reduce the complex structure of the object into a collection of simpler shapes and their geometric configuration.

These work have been influential for quite a few, quite a few years and then in the 80s David Lowe, here is another example of thinking how to reconstruct or recognize the visual world from simple world structures, this work is by David Lowe which he tries to recognize razors by constructing lines and edges and and mostly straight lines and their combination.

So there was a lot of effort in trying to think what what is the tasks in computer vision in the 60s 70s and 80s and frankly it was very hard to solve the problem of object recognition;

everything I've shown you so far are very audacious ambitious attempts but they remain at the level of toy examples or just a few examples.

Not a lot of progress have been made in terms of delivering something that can work in real world.

So as people think about what are the problems to solving vision one important question came around is: if object recognition is too hard, maybe we should first do object segmentation, that is the task of taking an image and group the pixels into meaningful areas.

We might not know the pixels that group together is called a person, but we can extract out all the pixels that belong to the person from its background; that is called image segmentation.

So here's one very early seminal work by Jitendra Malik and his student Jianbo Shi from Berkeley from using a graph theory algorithm for the problem of image segmentation.

Here's another problem that made some headway ahead of many other problems in computer vision, which is face detection.

Faces one of the most important objects to humans, probably the most important objects to humans, around the time of 1999 to 2000 machine learning techniques, especially statistical machine learning techniques start to gain momentum.

These are techniques such as support vector machines, boosting, graphical models, including the first wave of neural networks.

One particular work that made a lot of contribution was using AdaBoost algorithm to do real-time face detection by Paul Viola and Michael Jones and there's a lot to admire in this work.

It was done in 2001 when computer chips are still very very slow but they're able to do face detection in images in near-real-time and after the publication of this paper in five years time, 2006, Fujifilm rolled out the first digital camera that has a real-time face detector in the in the camera so it was a very rapid transfer from basic science research to real world application.

So as a field we continue to explore how we can do object recognition better so one of the very influential way of thinking in the late 90s til the first 10 years of 2000 is feature based object recognition and here is a seminal work by David Lowe called SIFT feature.

The idea is that to match and the entire object for example here is a stop sign to another stop sight is very difficult because there might be all kinds of changes due to camera angles, occlusion, viewpoint, lighting, and just the intrinsic variation of the object itself but it's inspired to observe that there are some parts of the object, some features, that tend to remain diagnostic and invariant to changes so the task of object recognition began with identifying these critical features on the object and then match the features to a similar object, that's a easier task than pattern matching the entire object.

So here is a figure from his paper where it shows that a handful, several dozen SIFT features from one stop sign are identified and matched to the SIFT features of another stop sign.

Using the same building block which is features, diagnostic features in images, we have as a field has made another step forward and start to recognizing holistic scenes.

Here is an example algorithm called Spatial Pyramid Matching; the idea is that there are features in the images that can give us clues about which type of scene it is, whether it's a landscape or a kitchen or a highway and so on and this particular work takes these features from different parts of the image and in different resolutions and put them together in a feature descriptor and then we do support vector machine algorithm on top of that.

Similarly a very similar work has gained momentum in human recognition so putting together these features well we have a number of work that looks at how we can compose human bodies in more realistic images and recognize them.

So one work is called the "histogram of gradients," another work is called "deformable part models," so as you can see as we move from the 60s 70s 80s towards the first decade of the 21st century one thing is changing and that's the quality of the pictures were no longer, with the Internet the the the growth of the Internet the digital cameras were having better and better data to study computer vision.

So one of the outcome in the early 2000s is that the field of computer vision has defined a very important building block problem to solve.

It's not the only problem to solve but in terms of recognition this is a very important problem to solve which is object recognition.

I talked about object recognition all along but in the early 2000s we began to have benchmark data set that can enable us to measure the progress of object recognition.

One of the most influential benchmark data set is called PASCAL Visual Object Challenge, and it's a data set composed of 20 object classes, three of them are shown here: train, airplane, person; I think it also has cows, bottles, cats, and so on; and the data set is composed of several thousand to ten thousand images per category and then the field different groups develop algorithm to test against the testing set and see how we have made progress.

So here is a figure that shows from year 2007 to year 2012.

The performance on detecting objects the 20 object in this image in a in a benchmark data set has steadily increased.

So there was a lot of progress made.

Around that time a group of us from Princeton to Stanford also began to ask a harder question to ourselves as well as our field which is: are we ready to recognize every object or most of the object in the world.

It's also motivated by an observation that is rooted in machine learning which is that most of the machine learning algorithms it doesn't matter if it's graphical model, or support vector machine, or AdaBoost, is very likely to overfit in the training process and part of the problem is visual data is very complex because it's complex our models tend to have a high dimension a high dimension of input and have to have a lot of parameters to fit and when we don't have enough training data overfitting happens very fast and then we cannot generalize very well.

So motivated by this dual reason, one is just want to recognize the world of all the objects, the other one is to come back the machine learning overcome the the machine learning bottleneck of overfitting, we began this project called ImageNet.

We wanted to put together the largest possible dataset of all the pictures we can find, the world of objects, and use that for training as well as for benchmarking.

So it was a project that took us about three years, lots of hard work; it basically began with downloading billions of images from the internet organized by the dictionary we called WordNet which is tens of thousands of object classes and then we have to use some clever crowd engineering trick a method using Amazon Mechanical Turk platform to sort, clean, label each of the images.

The end result is a ImageNet of almost 15 million or 40 million plus images organized in twenty-two thousand categories of objects and scenes and this is the gigantic, probably the biggest dataset produced in the field of AI at that time and it began to push forward the algorithm development of object recognition into another phase.

Especially important is how to benchmark the progress so starting 2009 the ImageNet team rolled out an international challenge called ImageNet Large-Scale Visual Recognition Challenge and for this challenge we put together a more stringent test set of 1.

4 million objects across 1,000 object classes and this is to test the image classification recognition results for the computer vision algorithms.

So here's the example picture and if an algorithm can output 5 labels and and top five labels includes the correct object in this picture then we call this a success.

So here is a result summary of the ImageNet Challenge, of the image classification result from 2010 to 2015 so on x axis you see the years and the y axis you see the error rate.

So the good news is the error rate is steadily decreasing to the point by 2012 the error rate is so low is on par with what humans can do and here a human I mean a single Stanford PhD student who spend weeks doing this task as if he were a computer participating in the ImageNet Challenge.

So that's a lot of progress made even though we have not solved all the problems of object recognition which you'll learn about in this class but to go from an error rate that's unacceptable for real-world application all the way to on par being on par with humans in ImageNet challenge, the field took only a few years.

And one particular moment you should notice on this graph is the the year 2012.

In the first two years our error rate hovered around 25 percent but in 2012 the error rate was dropped more almost 10 percent to 16 percent even though now it's better but that drop was very significant and the winning algorithm of that year is a convolutional neural network model that beat all other algorithms around that time to win the ImageNet challenge and this is the focus of our whole course this quarter is to look at to have a deep dive into what convolutional neural network models are and another name for this is deep learning by by popular popular name now it's called deep learning and to look at what these models are what are the principles what are the good practices what are the recent progress of this model, but here is where the history was made is that we, around 2012 convolutional neural network model or deep learning models showed the tremendous capacity and ability in making a good progress in the field of computer vision along with several other sister fields like natural language processing and speech recognition.

So without further ado I'm going to hand the rest of the lecture to to Justin to talk about the overview of CS 231n.

Alright, thanks so much Fei-Fei.

I'll take it over from here.

So now I want to shift gears a little bit and talk a little bit more about this class CS231n.

So this class focuses on one of these most, so the primary focus of this class is this image classification problem which we previewed a little bit in the contex of the ImageNet Challenge.

So in image classification, again, the setup is that your algorithm looks at an image and then picks from among some fixed set of categories to classify that image.

And, this might seem like somewhat of a restrictive or artificial setup, but it's actual quite general.

And, this problem can be applied in many different settings both in industry and academia and many different places.

So for example, you could apply this to recognizing food or recognizing calories in food or recognizing different artworks, different product out in the world.

So this relatively basic tool of image classification is super useful on its own and could be applied all over the place for many different applications.

But, in this course, we're also going to talk about several other visual recognition problems that build upon many of the tools that we develop for the purpose of image classification.

We'll talk about other problems such as object detection or image captioning.

So the setup in object detection is a little bit different.

Rather than classifying an entire image as a cat or a dog or a horse or whatnot, instead we want to go in and draw bounding boxes and say that there is a dog here, and a cat here, and a car over in the background, and draw these boxes describing where objects are in the image.

We'll also talk about image captioning where given an image the system now needs to produce a natural language sentence describing the image.

It sounds like a really hard, complicated, and different problem, but we'll see that many of the tools that we develop in service of image classification will be reused in these other problems as well.

So we mentioned this before in the context of the ImageNet Challenge, but one of the things that's really driven the progress of the field in recent years has been this adoption of convolutional neural networks or CNNs or sometimes called convnets.

So if we look at the algorithms that have won the ImageNet Challenge for the last several years, in 2011 we see this method from Lin et al which is still hierarchical.

It consists of multiple layers.

So first we compute some features, next we compute some local invariances, some pooling, and go through several layers of processing, and then finally feed this resulting descriptor to a linear SVN.

What you'll notice here is that this is still hierarchical.

We're still detecting edges.

We're still having notions of invariance.

And, many of these intuitions will carry over into convnets.

But, the breakthrough moment was really in 2012 when Jeff Hinton's group in Toronto together with Alex Krizhevsky and Ilya Sutskever who were his PHD student at that time created this seven layer convolutional neural network now known as AlexNet, then called Supervision which just did very, very well in the ImageNet competition in 2012.

And, since then every year the winner of ImageNet has been a neural network.

And, the trend has been that these networks are getting deeper and deeper each year.

So AlexNet was a seven or eight layer neural network depending on how exactly you count things.

In 2015 we had these much deeper networks.

GoogleNet from Google and VGG, the VGG network from Oxford which was about 19 layers at that time.

And, then in 2015 it got really crazy and this paper came out from Microsoft Research Asia called Residual Networks which were 152 layers at that time.

And, since then it turns out you can get a little bit better if you go up to 200, but you run our of memory on your GPUs.

We'll get into all of that later, but the main takeaway here is that convolutional neural networks really had this breakthrough moment in 2012, and since then there's been a lot of effort focused in tuning and tweaking these algorithms to make them perform better and better on this problem of image classification.

And, throughout the rest of the quarter, we're going to really dive in deep, and you'll understand exactly how these different models work.

But, one point that's really important, it's true that the breakthrough moment for convolutional neural networks was in 2012 when these networks performed very well on the ImageNet Challenge, but they certainly weren't invented in 2012.

These algorithms had actually been around for quite a long time before that.

So one of the sort of foundational works in this area of convolutional neural networks was actually in the '90s from Jan LeCun and collaborators who at that time were at Bell Labs.

So in 1998 they build this convolutional neural network for recognizing digits.

They wanted to deploy this and wanted to be able to automatically recognize handwritten checks or addresses for the post office.

And, they built this convolutional neural network which could take in the pixels of an image and then classify either what digit it was or what letter it was or whatnot.

And, the structure of this network actually look pretty similar to the AlexNet architecture that was used in 2012.

Here we see that, you know, we're taking in these raw pixels.

We have many layers of convolution and sub-sampling, together with the so called fully connected layers.

All of which will be explained in much more detail later in the course.

But, if you just kind of look at these two pictures, they look pretty similar.

And, this architecture in 2012 has a lot of these architectural similarities that are shared with this network going back to the '90s.

So then the question you might ask is if these algorithms were around since the '90s, why have they only suddenly become popular in the last couple of years?

And, there's a couple really key innovations that happened that have changed since the '90s.

One is computation.

Thanks to Moore's law, we've gotten faster and faster computers every year.

And, this is kind of a coarse measure, but if you just look at the number of transistors that are on chips, then that has grown by several orders of magnitude between the '90s and today.

We've also had this advent of graphics processing units or GPUs which are super parallelizable and ended up being a perfect tool for really crunching these computationally intensive convolutional neural network models.

So just by having more compute available, it allowed researchers to explore with larger architectures and larger models, and in some cases, just increasing the model size, but still using these kind of classical approaches and classical algorithms tends to work quite well.

So this idea of increasing computation is super important in the history of deep learning.

I think the second key innovation that changed between now and the '90s was data.

So these algorithms are very hungry for data.

You need to feed them a lot of labeled images and labeled pixels for them to eventually work quite well.

And, in the '90s there just wasn't that much labeled data available.

This was, again, before tools like Mechanical Turk, before the internet was super, super widely used.

And, it was very difficult to collect large, varied datasets.

But, now in the 2010s with datasets like PASCAL and ImageNet, there existed these relatively large, high quality labeled datasets that were, again, orders and orders magnitude bigger than the dataset available in the '90s.

And, these much large datasets, again, allowed us to work with higher capacity models and train these models to actually work quite well on real world problems.

But, the critical takeaway here is that convolutional neural networks although they seem like this sort of fancy, new thing that's only popped up in the last couple of years, that's really not the case.

And, these class of algorithms have existed for quite a long time in their own right as well.

Another thing I'd like to point out in computer vision we're in the business of trying to build machines that can see like people.

And, people can actually do a lot of amazing things with their visual systems.

When you go around the world, you do a lot more than just drawing boxes around the objects and classifying things as cats or dogs.

Your visual system is much more powerful than that.

And, as we move forward in the field, I think there's still a ton of open challenges and open problems that we need to address.

And, we need to continue to develop our algorithms to do even better and tackle even more ambitious problems.

Some examples of this are going back to these older ideas in fact.

Things like semantic segmentation or perceptual grouping where rather than labeling the entire image, we want to understand for every pixel in the image what is it doing, what does it mean.

And, we'll revisit that idea a little bit later in the course.

There's definitely work going back to this idea of 3D understanding, of reconstructing the entire world, and that's still an unsolved problem I think.

There're just tons and tons of other tasks that you can imagine.

For example activity recognition, if I'm given a video of some person doing some activity, what's the best way to recognize that activity?

That's quite a challenging problem as well.

And, then as we move forward with things like augmented reality and virtual reality, and as new technologies and new types of sensors become available, I think we'll come up with a lot of new, interesting hard and challenging problems to tackle as a field.

So this is an example from some of my own work in the vision lab on this dataset called Visual Genome.

So here the idea is that we're trying to capture some of these intricacies in the real world.

Rather than maybe describing just boxes, maybe we should be describing images as these whole large graphs of semantically related concepts that encompass not just object identities but also object relationships, object attributes, actions that are occurring in the scene, and this type of representation might allow us to capture some of this richness of the visual world that's left on the table when we're using simple classification.

This is by no means a standard approach at this point, but just kind of giving you this sense that there's so much more that your visual system can do that is maybe not captured in this vanilla image classification setup.

I think another really interesting work that kind of points in this direction actually comes from Fei-Fei's grad school days when she was doing her PHD at Cal Tech with her advisors there.

In this setup, they had people, they stuck people, and they showed people this image for just half a second.

So they flashed this image in front of them for just a very short period of time, and even in this very, very rapid exposure to an image, people were able to write these long descriptive paragraphs giving a whole story of the image.

And, this is quite remarkable if you think about it that after just half a second of looking at this image, a person was able to say that this is some kind of a game or fight, two groups of men.

The man on the left is throwing something.

Outdoors because it seem like I have an impression of grass, and so on and so on.

And, you can imagine that if a person were to look even longer at this image, they could write probably a whole novel about who these people are, and why are they in this field playing this game.

They could go on and on and on roping in things from their external knowledge and their prior experience.

This is in some sense the holy grail of computer vision.

To sort of understand the story of an image in a very rich and deep way.

And, I think that despite the massive progress in the field that we've had over the past several years, we're still quite a long way from achieving this holy grail.

Another image that I think really exemplifies this idea actually comes, again, from Andrej Karpathy's blog is this amazing image.

Many of you smiled, many of you laughed.

I think this is a pretty funny image.

But, why is it a funny image?

Well we've got a man standing on a scale, and we know that people are kind of self conscious about their weight sometimes, and scales measure weight.

Then we've got this other guy behind him pushing his foot down on the scale, and we know that because of the way scales work that will cause him to have an inflated reading on the scale.

But, there's more.

We know that this person is not just any person.

This is actually Barack Obama who was at the time President of the United States, and we know that Presidents of the United States are supposed to be respectable politicians that are [laughing] probably not supposed to be playing jokes on their compatriots in this way.

We know that there's these people in the background that are laughing and smiling, and we know that that means that they're understanding something about the scene.

We have some understanding that they know that President Obama is this respectable guy who's looking at this other guy.

Like, this is crazy.

There's so much going on in this image.

And, our computer vision algorithms today are actually a long way I think from this true, deep understanding of images.

So I think that sort of despite the massive progress in the field, we really have a long way to go.

To me, that's really exciting as a researcher 'cause I think that we'll have just a lot of really exciting, cool problems to tackle moving forward.

So I hope at this point I've done a relatively good job to convince you that computer vision is really interesting.

It's really exciting.

It can be very useful.

It can go out and make the world a better place in various ways.

Computer vision could be applied in places like medical diagnosis and self-driving cars and robotics and all these different places.

In addition to sort of tying back to sort of this core idea of understanding human intelligence.

So to me, I think that computer vision is this fantastically amazing, interesting field, and I'm really glad that over the course of the quarter, we'll get to really dive in and dig into all these different details about how these algorithms are working these days.

That's sort of my pitch about computer vision and about the history of computer vision.

I don't know if there's any questions about this at this time.

Okay.

So then I want to talk a little bit more about the logistics of this class for the rest of the quarter.

So you might ask who are we?

So this class is taught by Fei-Fei Li who is a professor of computer science here at Standford who's my advisor and director of the Stanford Vision Lab and also the Stanford AI Lab.

The other two instructors are me, Justin Johnson, and Serena Yeung who is up here in the front.

We're both PHD students working under Fei-Fei on various computer vision problems.

We have an amazing teaching staff this year of 18 TAs so far.

Many of whom are sitting over here in the front.

These guys are really the unsung heroes behind the scenes making the course run smoothly, making sure everything happens well.

So be nice to them.

[laughing] I think I also should mention this is the third time we've taught this course, and it's the first time that Andrej Karpathy has not been an instructor in this course.

He was a very close friend of mine.

He's still alive.

He's okay, don't worry.

[laughing] But, he graduated, so he's actually here I think hanging around in the lecture hall.

A lot of the development and the history of this course is really due to him working on it with me over the last couple of years.

So I think you should be aware of that.

Also about logistics, probably the best way for keeping in touch with the course staff is through Piazza.

You should all go and signup right now.

Piazza is really our preferred method of communication with the class with the teaching staff.

If you have questions that you're afraid of being embarrassed about asking in front of your classmates, go ahead and ask anonymously even post private questions directly to the teaching staff.

So basically anything that you need should ideally go through Piazza.

We also have a staff mailing list, but we ask that this is mostly for sort of personal, confidential things that you don't want going on Piazza, or if you have something that's super confidential, super personal, then feel free to directly email me or Fei-Fei or Serena about that.

But, for the most part, most of your communication with the staff should be through Piazza.

We also have an optional textbook this year.

This is by no means required.

You can go through the course totally fine without it.

Everything will be self contained.

This is sort of exciting because it's maybe the first textbook about deep learning that got published earlier this year by E.

N.

Goodfellow, Yoshua Bengio, and Aaron Courville.

I put the Amazon link here in the slides.

You can get it if you want to, but also the whole content of the book is free online, so you don't even have to buy it if you don't want to.

So again, this is totally optional, but we'll probably be posting some readings throughout the quarter that give you an additional perspective on some of the material.

So our philosophy about this class is that you should really understand the deep mechanics of all of these algorithms.

You should understand at a very deep level exactly how these algorithms are working like what exactly is going on when you're stitching together these neural networks, how do these architectural decisions influence how the network is trained and tested and whatnot and all that.

And, throughout the course through the assignments, you'll be implementing your own convolutional neural networks from scratch in Python.

You'll be implementing the full forward and backward passes through these things, and by the end, you'll have implemented a whole convolutional neural network totally on your own.

I think that's really cool.

But, we also kind of practical, and we know that in most cases people are not writing these things from scratch, so we also want to give you a good introduction to some of the state of the art software tools that are used in practice for these things.

So we're going to talk about some of the state of the art software packages like Tensor Flow, Torch, [Py]Torch, all these other things.

And, I think you'll get some exposure to those on the homeworks and definitely through the course project as well.

Another note about this course is that it's very state of the art.

I think it's super exciting.

This is a very fast moving field.

As you saw, even these plots in the imaging challenge basically there's been a ton of progress since 2012, and like while I've been in grad school, the whole field is sort of transforming ever year.

And, that's super exciting and super encouraging.

But, what that means is that there's probably content that we'll cover this year that did not exist the last time that this course was taught last year.

I think that's super exciting, and that's one of my favorite parts about teaching this course is just roping in all these new scientific, hot off the presses stuff and being able to present it to you guys.

We're also sort of about fun.

So we're going to talk about some interesting maybe not so serious topics as well this quarter including image captioning is pretty fun where we can write descriptions about images.

But, we'll also cover some of these more artistic things like DeepDream here on the left where we can use neural networks to hallucinate these crazy, psychedelic images.

And, by the end of the course, you'll know how that works.

Or on the right, this idea of style transfer where we can take an image and render it in the style of famous artists like Picasso or Van Gogh or what not.

And again, by the end of the quarter, you'll see how this stuff works.

So the way the course works is we're going to have three problem sets.

The first problem set will hopefully be out by the end of the week.

We'll have an in class, written midterm exam.

And, a large portion of your grade will be the final course project where you'll work in teams of one to three and produce some amazing project that will blow everyone's minds.

We have a late policy, so you have seven late days that you're free to allocate among your different homeworks.

These are meant to cover things like minor illnesses or traveling or conferences or anything like that.

If you come to us at the end of the quarter and say that, "I suddenly have to give a presentation "at this conference.

" That's not going to be okay.

That's what your late days are for.

That being said, if you have some very extenuating circumstances, then do feel free to email the course staff if you have some extreme circumstances about that.

Finally, I want to make a note about the collaboration policy.

As Stanford students, you should all be aware of the honor code that governs the way that you should be collaborating and working together, and we take this very seriously.

We encourage you to think very carefully about how you're collaborating and making sure it's within the bounds of the honor code.

So in terms of prerequisites, I think the most important is probably a deep familiarity with Python because all of the programming assignments will be in Python.

Some familiarity with C or C++ would be useful.

You will probably not be writing any C or C++ in this course, but as you're browsing through the source code of these various software packages, being able to read C++ code at least is very useful for understanding how these packages work.

We also assume that you know what calculus is, you know how to take derivatives all that sort of stuff.

We assume some linear algebra.

That you know what matrices are and how to multiply them and stuff like that.

We can't be teaching you how to take like derivatives and stuff.

We also assume a little bit of knowledge coming in of computer vision maybe at the level of CS131 or 231a.

If you have taken those courses before, you'll be fine.

If you haven't, I think you'll be okay in this class, but you might have a tiny bit of catching up to do.

But, I think you'll probably be okay.

Those are not super strict prerequisites.

We also assume a little bit of background knowledge about machine learning maybe at the level of CS229.

But again, I think really important, key fundamental machine learning concepts we'll reintroduce as they come up and become important.

But, that being said, a familiarity with these things will be helpful going forward.

So we have a course website.

Go check it out.

There's a lot of information and links and syllabus and all that.

I think that's all that I really want to cover today.

And, then later this week on Thursday, we'll really dive into our first learning algorithm and start diving into the details of these things.