**AI image recognition – exploring limitations and bias**

A hands-on activity to practise training and testing an artificial intelligence (AI) model, using cartoon faces, including a discussion about sources of potential algorithmic bias and how to respond to these sources.

**Year levels**: 5–6, 7–8



*Image:* [*Donna Burton/U.S. Customs & Border Control/Flickr*](https://www.flickr.com/photos/cbpphotos/26902922288)

# Suggested steps

If you already have a teacher account and student logins fully set up on the website Machine Learning for Kids, please skip to [Training the AI](#bookmark=id.1fob9te).

## Before you begin

This activity uses the online resource [Machine Learning for Kids](https://machinelearningforkids.co.uk). To enable your class to train and test an AI model so that it can recognise certain kinds (or features of) images, a class teacher must sign up on Machine Learning for Kids, then also sign up and obtain a couple of codes from another site (IBM Cloud). See the [video](https://publish.viostream.com/player/download/bxixurbns8keop) or steps below for a complete walkthrough, *current as of November 2019*.

1. Proceed to the login page at [machinelearningforkids.co.uk/#!/login](https://machinelearningforkids.co.uk/#!/login).
2. Select **Sign up**.
3. Select **A parent, teacher or leader of a coding club**.
4. Under ‘Create an unmanaged class account’, select **Sign up**.
5. Enter a username, your email address and intended use, then tick the checkbox.
6. Read the information at the bottom of same page in the set-up – Create an unmanaged class account). Here is a brief summary.  
     
   After signing up for Machine Learning for Kids, you, the teacher, will be required to set up an account on IBM Cloud in order to obtain access credentials and codes called ‘API keys’. You will not need to provide credit card information to use IBM Cloud, and free usage should be adequate for your class.
7. Select **CREATE CLASS ACCOUNT** at the bottom right of the same page.
8. *Note down the password* generated for you at the top of the page (and keep it in a safe place).
9. Check your email inbox for an email from Machine Learning for Kids. Click the link in the email to verify your account.
10. Return to the Machine Learning for Kids [login page](https://machinelearningforkids.co.uk/#!/login) and **Log in** with your username and password.
11. Select **Teacher** in the top menu to show the Administration page.
12. A pink box will indicate that the API Keys from IBM have not been obtained yet. Select **Watson API Keys** to see where these will need to be added.
13. From this point, you’ll need to follow the first page of steps in a [PDF](https://github.com/IBM/taxinomitis-docs/raw/master/docs/pdf/machinelearningforkids-ibmer.pdf) available on the Machine Learning for Kids [help page](https://machinelearningforkids.co.uk/#!/help) (Unmanaged class accounts). These steps include:
    * creating an IBM Cloud account
    * creating Watson Assistant API Keys and adding them to Machine Learning for Kids
    * creating Watson Visual Recognition API Keys and adding them to Machine Learning for Kids
    * setting up names and passwords for your students on Machine Learning for Kids (up to 30 users in total, including yourself).

Remember that you do *not* need to provide credit card information to use IBM Cloud, and free usage should be adequate for your class to do both this activity and others.

# Training the AI

Follow the steps below to train an AI model to recognise cartoon faces wearing glasses or sunglasses.

1. Log in at [machinelearningforkids.co.uk/#!/login](https://machinelearningforkids.co.uk/#!/login). Students use the names and passwords already set up by the teacher.
2. Go to **Projects**.
3. Select **Add a new project**.
4. Enter a project name, eg Algorithmic bias.
5. For Recognising ..., choose **images**.
6. Select the **Create** button.
7. From the ‘Projects’ page, select the project you just made.
8. Select **Train**.
9. Add three labels: glasses, sunglasses or noGlasses. These are the buckets you'll be putting the images in.
10. In a separate browser window, open [this gallery of training faces](https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/integrating-digital-technologies/ai-image-recognition-training-images/) (Cartoon set © Google LLC, [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).
11. With the windows side-by-side, drag the images into the appropriate bucket: glasses, sunglasses, or noGlasses. You should end up with 14 images in each bucket.   
    *Note, dragging does not work in Microsoft Edge (as of September 2019).*
12. When done, select **< Back to project**.
13. Select **Learn & Test**.
14. Select **Train new machine learning model**. This may take from 10 to15 minutes. The page will update when it's done.

IMPORTANT NOTE – AUTOMATIC DELETION OF AI MODELS: By default, after 24 hours Machine Learning for Kids automatically deletes AI models trained by students. As a teacher, you can increase this time to as much as 1.5 weeks (but remember that there is a limited number of models that a class can have at any time). Student training data is not deleted, so models can always be retrained.

# Testing the AI

Follow the steps below to test your AI model.

1. Once training is done, a box will appear on the same page to **Test with a web address for an image on the Internet**. [This gallery of test faces](https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/integrating-digital-technologies/ai-image-recognition-testing-images/) (Cartoon set © Google LLC, [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)) has more diverse skin colours than in the training dataset. Right-click on one face and copy the image URL, then paste it into that box. Then press the button to test it.
2. Now we’ll try out our model within a Scratch program. Select **< Back to project**.
3. Select **Make**.
4. Select **Scratch 3**. (You can also try Python.)
5. Select **Open in Scratch 3**.
6. To test an image, first change the sprite's current costume to the face you want to test. Download one of the images from [the test gallery](https://ibb.co/album/mkqCWF), then upload it as a costume for the sprite.
7. Code a program using the extra **Machine Learning for Kids** blocks and **Images** blocks. The screenshot below shows a sample program.

IMPORTANT NOTE – SAVING PROGRAMS: To accommodate each AI model, a *custom* Scratch 3 environment is launched when you select **Make**. This is different from the regular Scratch 3 environment accessed at scratch.mit.edu. Programs cannot be shared or uploaded across the two different environments. This means several things:

* Students cannot easily share their programs with the teacher or other students. Assessment evidence may need to be screenshots, video recordings or teacher observations.
* Students who wish to keep a program at the end of a lesson must download it as a file. Later, they must re-enter the custom Scratch environment by clicking the **Make** button in their Machine Learning for Kids project. Then, they can upload the program file again. Also see the note Automatic deletion of models ‘at the end of Training the AI’

[](http://www.digitaltechnologieshub.edu.au/images/default-source/dt/ai-image-recognition/sample-scratch-screenshot_placeholder.png)

*Image: Sample screenshot showing a very simple program within a Scratch 3 custom environment using* ***Machine Learning for Kids*** *blocks and* ***Images*** *blocks. Click on image to open a high-resolution version*.

# Discussion

* In both image sets, the faces are deliberately varied in size and placed in varied positions. Do you think this reflects a real-life scenario such as passport photo checking? Why / why not?
* Do you think real faces would be easier or more difficult for the system than these cartoon faces? What sort of variations occur with real photographs?
* Try testing your AI model with several images from the test gallery, which contains faces with more diverse skin colours than the images in the training gallery.
  + Did you notice any difference in the accuracy of the model’s system when faces with more diverse skin colours were tested? Did the system get it right and, if so, with how much confidence?
  + If you did find a discrepancy, what technical reasons could you give for why this occurred? (See **Why is this relevant?** below for possible reasons.)
* When a computer system creates unfair outcomes, this is often referred to as algorithmic bias (go to **Why is this relevant?** for more information). If a digital solution has more difficulty distinguishing faces for particular ethnic groups, can you think of a real-world situation where this might cause unfair outcomes?
  + Hint: facial recognition technology is in some cases already being used to prove identity. Search on ‘facial recognition identity examples’.
* Can you think of any proven real-world examples of algorithmic bias?
  + One past example is the [Nikon camera controversy](http://content.time.com/time/business/article/0,8599,1954643,00.html) in 2009–10, when an algorithm designed to detect if photograph subjects were blinking mis-interpreted a number of Asian subjects as having their eyes closed.

# Extensions

* The sample screenshot of the Scratch program in ‘Testing the AI’ merely displays the output from the machine learning model. Expand the program into an application – for example, a hypothetical passport checking system that rejects faces that do not meet minimum ‘confidence’ requirements.
* For secondary students, try creating a Python program as an alternative to Scratch.
* Build a gallery of faces (cartoon or real) and try making your own AI model.

***NOTE: For privacy reasons, it is recommended that photos do not include student faces or other personal identifiers.***

# Why is this relevant?

Algorithmic bias creates errors that may lead to unfair or dangerous outcomes, for instance, for one or more groups of people, organisations, living things and the environment.

Algorithmic bias is often unintentional. It can arise in several ways. Some examples:

1. As a result of **design**. The software’s algorithms (the instructions and procedures used to make decisions), may have been coded based on incorrect assumptions, outdated understandings, prioritised motives or even technical limitations. Or the design may simply be misapplied, used for a purpose for which it was never intended.
   * In the case of our facial recognition example, such systems may have difficulty recognising the outlines of faces with dark skin colour against a dark background because of the algorithm’s dependence on distinguishing sufficient light contrast. The cartoons in our activity frequently use black lines for face outlines as well as for the outlines of glasses.
2. Due to inadequate or biased **data**. Programmers have long been familiar with Garbage In, Garbage Out (GIGO), meaning that poor quality data used as input to a computer system will tend to result in poor quality output and decisions. Machine learning systems are trained from data sets of text, images or sounds, and these may be restricted or unrepresentative.
   * In the case of our facial recognition example, a system may be trained on an insufficiently diverse data set – for example, one based predominantly on faces of light skin colour or facial features associated with a limited range of ethnic groups.

# Assessment

## Teacher assessment

Using original photographs created by students, or content sourced online, students work in pairs to create a new AI models for one of the example topics below:

* Distinguish different sign language hand postures
* Distinguish different semaphore poses
* Distinguish favourite sport team based on face paint
* A topic chosen by a student or students.

Choose from the following suggested assessment approaches and task that will best suit your students.

|  |  |
| --- | --- |
| **Possible tasks** | **Relevant content descriptor(s)** |
| Run a two-part investigation by training the model using a set of images that is ‘biased’ or skewed in some way before the first test, then retraining the model using a set of images that is ‘unbiased’ (for the purposes of the investigation) before the second test.  Write a report, detailing:   * biased image set * unbiased image set * testing image set * results of both tests, including decisions and confidence figures * an evaluation discussing any difference in results between the biased and unbiased models. | [ACTDIP016](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP016) / [ACTDIP026](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP026)  [ACTDIP017](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP017) / [ACTDIP027](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP027)  [ACTDIP021](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP021) / [ACTDIP031](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP031) |
| Design and develop a Scratch or Python program to simulate a computer application that uses your AI model.  Create a ‘quick start’ guide or video to demonstrate and explain the program.  *IMPORTANT NOTE: It is difficult for students to share finished programs from Machine Learning for Kids. See the note SAVING PROGRAMS in the Testing the AI section.* | [ACTDIP016](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP016) / [ACTDIP026](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP026)  [ACTDIP017](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP017) / [ACTDIP027](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP027)  [ACTDIP020](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP020) (Scratch) / [ACTDIP030](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP030) (Python)  [ACTDIP021](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP021) / [ACTDIP031](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP031) |
| Write an essay, with reference to real examples, about what software developers could do to avoid or reduce algorithmic bias in the future. | [ACTDIP016](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP016) / [ACTDIP026](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP026)  [ACTDIP017](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP017) / [ACTDIP027](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP027)  [ACTDIP021](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP021) / [ACTDIP031](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP031) |

# Australian Curriculum alignment

## Technologies – Digital Technologies

|  |  |  |
| --- | --- | --- |
| **Strand** | **Year** | **Content Description** |
| Processes and Production Skills | Years 5–6 | Acquire, store and validate different types of data, and use a range of software to interpret and visualise data to create information ([ACTDIP016](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP016))  Define problems in terms of data and functional requirements drawing on previously solved problems ([ACTDIP017](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP017))  Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input ([ACTDIP020](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP020))  Explain how student solutions and existing information systems are sustainable and meet current and future local community needs ([ACTDIP021](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP021)) |
| Processes and Production Skills | Years 7–8 | Analyse and visualise data using a range of software to create information, and use structured data to model objects or events ([ACTDIP026](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP026))  Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints ([ACTDIP027](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP027))  Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language ([ACTDIP030](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP030))  Evaluate how student solutions and existing information systems meet needs, are innovative, and take account of future risks and sustainability ([ACTDIP031](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP031)) |

## ICT Capability

Typically, by the end of Year 8, students:

**Identify the impacts of ICT in society** (Applying social and ethical protocols and practices when using ICT)

* explain the benefits and risks of the use of ICT for particular people in work and home environments

**Select and evaluate data and information** (Investigating with ICT)

* assess the suitability of data or information using appropriate own criteria