**Smartphone security**

This lesson provides an opportunity to investigate security measures, including those powered by artificial intelligence (AI), that are used to protect users from unauthorised (unapproved, unwanted) access to their digital devices.

Unauthorised use of our digital devices is an everyday threat and is a key aspect of cybersecurity. We need to ensure measures are in place to protect our personal information and privacy.

There are a range of security measures used, depending on the device. On a smartphone, AI has opened up the way to use biometrics (for example, thumbprint or face recognition) to protect the authorised user of a phone and deny access to unauthorised users.

Students can create computer programs to demonstrate a security measure such as using a Personal Identification Number (PIN) – for example, a 4-digit code. Alternatively, students can incorporate a type of image recognition to mimic AI. Another option is to use an AI tool to create a model and incorporate this into their Scratch program.

This lesson provides a level of differentiation to cater for students’ range of programming skills.

This set of lesson ideas was developed in collaboration with the [Digital Technologies Institute](https://www.digital-technologies.institute/)**.**

**Year levels**: 3–4, 5–6

# Preliminary notes

Use Scratch 3.0 or a similar visual programming environment to create a digital solution.

Use [Teachable Machine to explore image recognition using an AI tool.](https://teachablemachine.withgoogle.com/)**[Note](https://teachablemachine.withgoogle.com/)**[: This tool requires an internet connection and access to the device’s camera.](https://teachablemachine.withgoogle.com/)

Use [Machine Learning for Kids](https://machinelearningforkids.co.uk/) to incorporate AI into the digital solution.
**Note**: Before the activity, this tool requires an initial set up by the teacher completed prior to the lesson. View the step-by-step guide to help with this process.

The following matrix shows which tools are required for each level in the **Plugged** part of this lesson.

|  |  |  |
| --- | --- | --- |
| **Level** | **Tools required** | **Artificial intelligence used** |
| Easy | Teachable Machine | Yes |
| Medium | Scratch 3.0 | No |
| Medium to high | Scratch, Machine Learning for Kids | Yes |

### <H3>Machine learning for Kids set-up

### This set-up needs to be done by the teacher only once before students can start their projects. It is only required for the activity at the ‘Medium to high’ level.

This activity uses the online resource [Machine Learning for Kids](https://machinelearningforkids.co.uk). To enable your class to program for the recognition of images, you as 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 or steps below for a complete walkthrough, *current as of November 2019*.

View the step-by-step instructions on this set-up [video](https://publish.viostream.com/play/bxixurbni45ad8).

1. Proceed to the login page at [machinelearningforkids.co.uk/#!/login](https://machinelearningforkids.co.uk/#!/login).
2. Select **Sign up**.
3. Choose **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 the 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. Select 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. You can have 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 this activity and others.

# Suggested steps

## Unplugged activities

1. Discuss the many things that people use their smartphones for in daily life. Then ask: ‘How do users make sure their smartphone is secure?’

List security measures used to protect a user from someone else’s unwanted (unauthorised) use of their smartphone.

Examples include:

* Personal identification number (PIN)
* thumbprint or fingerprint scan and recognition
* eye (iris) scanning and recognition.

Discuss which of these would use AI to perform the security measure. Explain that biometric security often uses image recognition to reform this function.

2. Ask students to list or draw the steps they or other people follow to unlock their own smartphones. *(Bear in mind that many primary-age students may not have their own smartphone.)* Consider a flowchart or similar as a suitable representation of the process. To encourage thinking, pose the following questions:

* 1. How do users access their smartphone?
	2. What prompts might people see on screen to unlock their device?
	3. How does the user gain authorised access?
	4. What happens when access is granted?
	5. What happens when access is denied?

## Plugged activities

These activities explore ways of implementing a digital solution that demonstrates a smartphone security measure.

### Use an AI image recognition tool (skill level: easy)

Use the AI tool [Teachable Machine](https://teachablemachine.withgoogle.com/) to create a simple demonstration of how facial recognition can be used to unlock a smartphone. No coding is required in this activity.

**Note**: This tool requires an internet connection and access to the device’s camera. View the tutorial first as a class, to make sure students are clear on how the tool works. The AI needs to be trained on data, in this case, each student’s image.



Image: sample screen fromthe AI tool Teachable Machine v 1, showing INPUT feeding into the LEARNING component, the elements of which lead to an OUTPUT (with the GIF option selected, Sound and Speech, being the other two tabs).

1. In groups of three, students choose one person to be the ‘owner’ that the AI will recognise and grant access to, allowing that person to unlock the phone. The other two students will be recognised by the AI as not being the owner, so they will be denied access.
2. Each student will train the AI using their image captured by the device’s camera shown on screen as the INPUT. For example, the student selected as the owner can train the AI first; this is the TRAIN GREEN section. The student captures at least 10 images of themselves as training data. They then change the output from the GIF cat image to select ‘Speech’ and type in ‘Granted’.
3. The other two students follow the same process but change the output in ‘Speech’ to ‘Denied’).
4. Students record a video of their security measure and explain both the way it works and how well it works.
5. Ask your students to consider these questions:
	1. How much data ( in the form of images of each person) is needed to make sure the AI works successfully?
	2. Did you capture your image behind different backgrounds? Why might this be important for the AI to work well?

### Data bias experiment

To explore data bias when training an AI, refer to this lesson plan. It encourages students to experiment with changing the background of images when training an AI.

### Program a smartphone to unlock with a PIN (skill level: medium)

This option uses Scratch 3.0 or a similar programming environment. It requires the following programming skills:

1. Create a sprite (or upload an image to represent a smartphone).
2. Change the costumes of the sprite smartphone (to represent access denied and access granted).
3. Use the ‘When this sprite is clicked’ block.
4. Use the ‘Broadcast message’ and ‘When I receive’ message blocks.
5. Use the ‘ask’ block to enable a user to input an answer (in this case a 4-digit pin).
6. Use the ‘if … then’ and ‘else’ block to decide when to grant or deny access.
7. *Optional*: use a ‘forever’ block to loop the program.

Present the challenge of creating a program in Scratch 3.0 (or other familiar visual programming language) that uses a PIN to unlock a smartphone.

Depending on your students’ skill level, choose from among these:

* Provide students with the sample completed Scratch program [Phone unlock/lock](https://scratch.mit.edu/projects/340738632/fullscreen/) to play with and use to remix their own version.
* Provide students with the basic code of inputting four numbers to change screens, using the sample Scratch program [Basic Phone lock/unlock](https://scratch.mit.edu/projects/340799447/editor). Students then add further screens and functionality.
* Ask students to design their own digital solution, creating an algorithm that they follow to implement their solution.

A basic program can use the logic of first asking for a PIN – personal identification number – as an ‘answer’. An ‘ask’ block is selected for this purpose. It can be initiated when the green flag is clicked/selected.



Image: Screen capture of Scratch 3 Phone lock/unlock: ask for PIN (pin)

If the PIN (answer) is correct, broadcast ‘Unlock’ and switch the costume.

**Note**: The ‘costume’ changes depending on the broadcast message.



Image: Screen capture of Scratch 3-coded program showing broadcast message ‘Unlock’ (access granted), with matching costume.

If the PIN is wrong (an ‘else’ situation), broadcast ‘Locked’, say ‘Incorrect pin. Try again’, and switch the costume to a locked one.



Image: Screen capture of Scratch 3-coded program, showing broadcast message ‘Locked’ (access denied), with matching costume.

A flowchart representation of the steps described above could look like this:



Image: Flowchart representation of ‘Lock’/’Unlock’ steps depending on whether a PIN is correct or not.

Sample code: [Basic Phone lock/unlock](https://scratch.mit.edu/projects/340799447/editor)

#### Extension

A more complex solution can include multiple sprites and an option to access the screen by selecting a **home** button. This will involve showing and hiding sprites at different stages of the program. A forever block can be used to animate the home button by looping the next costume block.



Image: Screen capture of Scratch 3.0-coded program where **Home** button is used as one step to lock or unlock a smartphone



Image: Screen capture of Scratch 3.0-coded program with broadcast message ‘Unlock’

Sample completed program: [Phone unlock/lock](https://scratch.mit.edu/projects/340738632/fullscreen/)

(This has home screen with animated home button)

#### Ideas for a remix

Students:

* change the value of the answer to a different pin number or string
* create their own costumes for the smartphone
* add a block of program code that locks the screen after a certain number of failed attempts
* create supplementary screens that will appear after access.

**Program a smartphone to unlock with an image
(skill level: medium to high)**

The first solution involves using Scratch 3.0.

The second option requires a higher level of skill, but students learn how to train an AI and incorporate the AI model into their Scratch programming. This option requires the use of an AI tool: Machine Learning for Kids.

**Coding with Scratch 3.0**

The first step is to have a Start screen that asks the user to scan their facial image. A sprite can be made and labelled ‘Scan’.



Image: Screen capture of Start screen coded in Scratch3.0

In the following example, avatars have been used. Students use the broadcast message to switch the background to show ‘Scanning’.



Image: Screen capture of ‘Scanning’ image coded in Scratch3.0

Students label their own image as *granted* and identify the costume number. In our program, **If** costume number 2 is scanned, **then** broadcast is granted and the screen will show granted. **Else** any other image (costume number will be denied).



Image: Screen capture of granted image coded in Scratch3.0



Image: Screen capture of denied image coded in Scratch3.0

**Ideas for a remix**

Students can:

* add their own images (upload a photo or take a photo using a device camera)
* add more than two images (as costumes)
* create their own Start screen
* create a program based on their own algorithm
* design the user interface.

Sample code: Scratch [Basic Phone lock/unlock IMAGE](https://scratch.mit.edu/projects/341033989/editor/)

**<H4>Coding with Scratch 3.0 and incorporating an AI model**

Visit the [worksheet](https://machinelearningforkids.co.uk/#!/worksheets) page on Machine Learning for Kids. Select the project Face Lock. Download the step-by-step guide, which has explanations and colour screenshots for students to follow.
**Note**: *Although the guide is labelled ‘easy’ on their site, relative to our activities it is medium to high.*

Here is the main script students would use. The main difference between this and the code above (see Sample code: Scratch Basic Phone lock/unlock IMAGE) is that the AI is invoked to determine if an image has been recognised. The screenshot below shows the additional black Machine Learning for Kids blocks.



Image: AI model incorporated into Scratch 3.0

Discuss the ways the AI responded to the different facial images.

1. How well did the AI recognise the images?
2. What did you learn about the training of an AI?
3. How important is the background when training and recognising images?

# Discussion

Share what you have learned about AI and how ‘smart’ a computer can be.

* Consider the program you created. How well did it use or mimic AI?

# Why is this relevant?

Algorithms and programming are essential to developing machines powered by artificial intelligence (AI). AI is the ability of machines to mimic human capabilities in a way that we would consider 'smart'.

In conventional programming the computer is provided with a set of instructions for a defined set of scenarios. In the 4-digit PIN program and the basic image unlock program, the students hard-coded the program with specific inputs to create an output. To include an AI model we used Machine Learning for Kids to use the AI to recognise images as ‘granted’ or ‘denied’.

Machine learning (ML) is an application of AI. With machine learning, we give the machine lots of examples of data, demonstrating what we would like it to do so that it can figure out how to achieve a goal on its own. The machine learns and adapts its strategy to achieve this goal.

This lesson focuses on:

* text recognition
* image recognition
* training an AI with suitable data.

# Resources

AI tool [Teachable Machine](https://teachablemachine.withgoogle.com/)

Visual programing environment [Scratch 3.0](https://scratch.mit.edu)

AI tool that enables students to train their own AI model and incorporate into Scratch 3.0

[Machine Learning for Kids](https://machinelearningforkids.co.uk/)

Articles:

[Iris Recognition](https://www.eff.org/pages/iris-recognition)  https://www.eff.org/pages/iris-recognition

Iris scans <https://www.explainthatstuff.com/how-iris-scans-work.html>

# Assessment

## Teacher assessment

Choose from these suggested assessment approaches and tasks those that will best suit your students.

|  |  |  |
| --- | --- | --- |
| **Relevant task** UnpluggedPlugged: EasyPlugged: MediumPlugged: Medium to high | **Possible tasks** | **Relevant content descriptor(s)** |
| ALL | Describe the data required and stored in security measures used to protect from unauthorised use of a digital device.  | (ACTDIK008)/(ACTDIK015) |
| Plugged: EasyPlugged: MediumPlugged: Medium to high | Describe the hardware and software used in making your program that is used to protect the user from unauthorised use of their smartphone. Draw a representation of the algorithm. Explain how it works. What data is used and stored?  | ACTDIK007 / ACTDIK014(ACTDIP010) / (ACTDIP018) |
| Plugged: Medium to high | For those that created an AI model using Machine Learning for Kids, draw a labelled diagram of the AI in action. Identify (label) your inputs and the AI outputs. * Describe how the AI responds to different images.
 | [(ACTDIK014)](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACTDIK001) / [(ACTDIK023)](http://www.scootle.edu.au/ec/search?accContentId=ACTDIK007)(ACTDIP012) / (ACTDIP021) |
| ALL  | This type of AI uses image recognition. How is this type of AI used in our daily lives? List some examples. How would you prefer this type of AI be used? What issues might arise?  | (ACTDIP021) / (ACTDIP031) |

# Australian Curriculum Alignment

## Technologies – Digital Technologies

**Years 3–4**

Identify and explore a range of digital systems with peripheral devices for different purposes, and transmit different types of data (ACTDIK007)

Recognise different types of data and explore how the same data can be represented in different ways (ACTDIK008)

Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them (ACTDIP010)

Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input (ACTDIP011)

Explain how student solutions and existing information systems meet common personal, school or community needs (ACTDIP012)

Plan, create and communicate ideas and information independently and with others, applying agreed ethical and social protocols (ACTDIP013)

**Years 5–6**

Examine the main components of common digital systems and how they may connect together to form networks to transmit data (ACTDIK014)

Examine how whole numbers are used to represent all data in digital systems (ACTDIK015)

Design a user interface for a digital system (ACTDIP018)

Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) (ACTDIP019)

Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input (ACTDIP020)

Explain how student solutions and existing information systems are sustainable and meet current and future local community needs (ACTDIP021)

Plan, create and communicate ideas and information, including collaboratively online, applying agreed ethical, social and technical protocols (ACTDIP022)

## ICT Capability

Typically, by the end of Year 4, students:

**Apply digital information security practices**

Independently apply standard guidelines and techniques for particular digital systems to secure digital information

**Apply personal security protocols**

Apply standard guidelines and take action to avoid the common dangers to personal security when using ICT and apply appropriate basic social protocols when using ICT to communicate with unknown audiences

Typically, by the end of Year 6, students:

**Apply digital information security practices**

Independently apply strategies for determining and protecting the security of digital information and assess the risks associated with online environments

**Apply personal security protocols**

Identify the risks to identity, privacy and emotional safety for themselves when using ICT and apply generally accepted social protocols when sharing information in online environments, taking into account different social and cultural contexts