

# Pi - DETECT

USER MANUAL

# 1. QUICKSTART

## 1. CHANGE INSTAVULT DEFAULT KEY

- 1.1. The first thing to do in order to properly configure the application is to change the default key with you personal, **private key** that will only be known to you.
- 1.2. Run `sh insta-vault`, the software will automatically guide you to create your fist private key. The first old key is the default key, **ddm-01**.

```

pi@raspberrypi:~$ sh insta-vault
INSTA-VAULT V1.0 (c) 2022 - G.PASQUA.DEV - g.pasqua.dev@hotmail.com

@insta-vault: LICENSE VALIDATED SUCCESSFULLY
@insta-vault: Device License: [ :64] 9GpF0/++rutqmFqyMIAbHBAFVsdMvefH5BuA9IXRXZZ28uYRQsAwJ5xEuztnBZok
@insta-vault: Client ID: [ :64] VUtIBkrZfiLbwGvmyACidhxyntTb1b+ Ae2gCkXrL3f2wWvYkTFom4iBNksNJLZs
@insta-vault: Root Path: /home/pi/Documents/apps/pibird/
@insta-vault: Order Hash: 55c2254bb97f54e8ea7aac4f87678ce5f82eef272ae606cb087ceef6bc1942bb
@insta-vault: Current Mode: permanent
@insta-vault: Max Login Cycles: 10
@insta-vault: Cycle Status: 0
@insta-vault: Custom PiBird Caption: 'Secured with InstaVault!'

@insta-vault#recordL2: Default L2 Key Detected. Redirecting to Initalization.

Input OLD L2_Key.. ddm-01
Input NEW L2_Key.. prova
@insta-vault#recordL2: User Key Updated. Data re-encrypted.
    
```

- 1.3. **Do NOT lose the keys.** Even if you will have to change it because you think it's compromise, you **cannot lose the old key** without losing access to strongly encrypted credentials that will lock or alter the behavior of the software. So, always write out both the old key and the current key you are using.

## 2. RECORD INSTAGRAM LOGIN DATA

- 2.1. Run `sh insta-vault -su username -sp password` to safely store your username and password in the vault file.

```

pi@raspberrypi:~$ sh insta-vault -su username -sp password
INSTA-VAULT V1.0 (c) 2022 - G.PASQUA.DEV - g.pasqua.dev@hotmail.com

@insta-vault: LICENSE VALIDATED SUCCESSFULLY
@insta-vault: Device License: [ :64] 9GpF0/++rutqmFqyMIAbHBAFVsdMvefH5BuA9IXRXZZ28uYRQsAwJ5xEuztnBZok
@insta-vault: Client ID: [ :64] VUtIBkrZfiLbwGvmyACidhxyntTb1b+ Ae2gCkXrL3f2wWvYkTFom4iBNksNJLZs
@insta-vault: Root Path: /home/pi/Documents/apps/pibird/
@insta-vault: Order Hash: 55c2254bb97f54e8ea7aac4f87678ce5f82eef272ae606cb087ceef6bc1942bb
@insta-vault: Current Mode: permanent
@insta-vault: Max Login Cycles: 10
@insta-vault: Cycle Status: 0
@insta-vault: Custom PiBird Caption: 'Secured with InstaVault!'

Setting Instagram Username..
@insta-vault#edit-settings: Username Updated.

USER CHANGED
Setting Instagram Password..
@insta-vault#edit-settings: Password Updated.

PASS CHANGED
    
```

## 3. RUN PIBIRD

- 3.1. You can already launch pibird through `sh pibird`, it will automatically pair with insta-vault and gather all the required data.

```

@pibird#display_settings: LOADED SETTINGS
@pibird#display_settings: PIR Sensor registered on GPIO: #4
@pibird#display_settings: Lens Adjusting time: 2s
@pibird#display_settings: Model Confidence Threshold: 40.0%
@pibird#display_settings: Model Resources Root Path: /home/pi/Documents/apps/pibird/
@pibird#display_settings: Dataset Shift: 2 positions
@pibird#display_settings: Camera Profile Setting: DDM-01
@pibird#display_settings: Instagram Enabled: True
@pibird#display_settings: Image Size: 600 px

@pibird: PIR Sensor on GPIO 4 waiting for motion..
@pibird: Motion detected. Running Recognition Model..
@pibird: Model output: {'bird-type': 'Common Raven', 'probability': '99.2%', 'insta-pass': True}
@pibird: Sending to Instagram..

FOUND: w:600 h:600 r:1.0
2022-08-23 16:42:18,286 - INFO - Photo '/home/pi/Documents/apps/pibird/images/bird.jpg' is uploaded.
@pibird#instabot_run: Posted successfully on Instagram.
@pibird: PIR Sensor on GPIO 4 waiting for motion..
    
```

- 3.2. If your insta-vault is set to 'manual', you will have to add an additional parameter and run pibird through the `sh pibird -l2 private_key` as explained in the next paragraphs.

## 2. INSTAVAUULT

### 1. WHAT'S INSTAVAUULT?

- 1.1. InstaVault will record and safely store your instagram user data. It has three modes of operation:
- **Permanent (L1 KEY):** This mode has the lowest safety but it's the most recommended for an autorun-at-startup setup. The software will use an `embedded key (L1 KEY)` derived from the license ID in order to encrypt the data.
  - **Cyclic (L1 + L2 KEY):** This mode adds a safety layer to the permanent mode, being a variant of it with limited login cycles. The maximum amount of login cycles can be setup by the user using the dedicated command, and cannot however exceed `99 cycles`. When the cycles are over, the account will switch to disabled and require the `L2 KEY` to be switched back to active.
  - **Manual (L1 + L2 KEY):** This mode is recommended for maximum safety with autorun-onstartup disabled. It will prompt for the `personal user key (L2 KEY)` each time it has to login. When switching to and from `manual mode` the `L2 KEY` will be asked several times in order to validate and re-encrypt at each given step.
  - **Disabled (L2 KEY):** The account is disabled when it violates the maximum allowed amount of login cycles, or can be disabled on purpose by the user to prevent the software giving login data when the `pibird tunnel` is opened.

### 2. DISPLAY CURRENT SETTINGS

2.1. `-ds :`

- **Displays the current status and settings of InstaVault.**
- **Unit:** *no argument* *(none)*
- **Example:** `-ds`

### 3. SET PERSONAL KEY (L2\_KEY)

3.1. `-sl2 :`

- **Sets a new private key. The software will ask for the previous key in order to re-encrypt the records with the new key.**
- **Unit:** *no argument* *(none)*
- **Example:** `-sl2`
- **Returns:**
  - `L2_KEY_CHANGED` if the key has been successfully saved.
  - `L2_OLD_KEY_MISMATCH` if the old key is wrong.
    - *You will have to setup again the username and password because of this security breach.*

#### 4. CHECK PERSONAL KEY (L2\_KEY)

##### 4.1. `-cl2:`

- **Checks the saved private key with the provided user input.**
- **Unit:** *no argument* (none)
- **Example:** `-cl2`
- **Returns:**
  - `L2_CHECK_OK` if the keys are matching.
  - `L2_CHECK_ERROR` if the keys are different.

#### 5. GET ACCOUNT STATE

##### 5.1. `-as:`

- **Checks whether the InstaVault account is enabled or disabled.**
- **Unit:** *no argument* (none)
- **Example:** `-as`
- **Returns:**
  - `True` if the account is enabled.
  - `False` if the account is disabled.

#### 6. SET IG USERNAME

##### 6.1. `-su username:`

- **Set the IG account username to be stored in the vault.**
- **Unit:** *any username* (string)
- **Example:** `-su myusername`
- **Returns:**
  - `USER_CHANGED` if the username is successfully saved.
  - `MISMATCH` if there is problem due to external tinkering with the saving process or with encryption / decryption process.

#### 7. SET IG PASSWORD

##### 7.1. `-sp password:`

- **Set the IG account password to be stored in the vault.**
- **Unit:** *any password* (string)
- **Example:** `-su mypassword`
- **Returns:**
  - `PASS_CHANGED` if the password is successfully saved.
  - `MISMATCH` if there is problem due to external tinkering with the saving process or with encryption / decryption process.

#### 8. SET LOGIN MODE

##### 8.1. `-sm mode:`

- **Set the IG account password to be stored in the vault.**
- **Unit:** *mode [ manual | permanent | cyclic ]* (string)
- **Example:** `-sm permanent` (default)
- **Returns:**
  - `MODE_CHANGED` if the account operation mode is successfully changed.
  - `MISMATCH` if there is problem due to external tinkering with the saving process or with encryption / decryption process.

## 9. SET MAX LOGIN CYCLES

### 9.1. `-smc cycles_amount:`

- **Set the InstaVault maximum allowed login attempts.**
- **Unit:** cycles-amount [1 to 99] *(integer)*
- **Example:** `-smc 10` *(default)*
- **Returns:**
  - `MAX_CYCLES_CHANGED` if the cycles amount is successfully saved.
  - `MISMATCH` if there is problem due to external tinkering with the saving process or with encryption / decryption process.

## 10. SET CUSTOM IG CAPTION

### 10.1. `-scc custom_caption:`

- **Set an IG custom caption to be displayed after the model result.**
- **Unit:** any string *(string)*
- **Example:** `-scc Secured with InstaVault!` *(default)*
- **Returns:**
  - `PIBIRD_CAPTION_CHANGED` if the custom caption is successfully saved.
  - `MISMATCH` if there is problem due to external tinkering with the saving process or with encryption / decryption process.

## 11. RESTORE DISABLED INSTAVULT ACCOUNT

### 11.1. `-recv`

- **Attempts to recover a disabled InstaVault account if the L2 KEY is correct.**
- **Unit:** no argument *(none)*
- **Example:** `-recv`
- **Returns:**
  - `ACCOUNT_RESTORED` if the L2 KEY is correct, the account will then be restored.
  - `L2_KEY_MISMATCH` if the L2 KEY is incorrect, the account will then remain disabled.
  - `NOTHING_TO_RESTORE` if the account is already enabled.

## 12. OPEN PIBIRD PRIVATE TUNNEL

### 12.1. `-pibird`

- **Opens a private tunnel with PiBird. If manual mode is enabled, you will have to pass pibird the -l2 parameter with your private key to login; else both applications will automatically pair.**
- **Unit:** no argument *(none)*
- **Example:** `-pibird`
- **Returns:**
  - `PERMANENT_LOGIN_OK`, `CYCLIC_LOGIN_OK`, `MANUAL_LOGIN_OK` if the credentials are correctly passed for each selected mode.
  - `MISMATCH_L2_SAVED_KEY` if the passed L2 KEY is wrong and manual login fails.

### 3. PIBIRD

#### 1. SET PIR SENSOR HARDWARE ADDRESS

##### 1.1. `-pin pin_number:`

- Allows the user to select a GPIO number for the PIR Sensor. You can also run a separated PIR Sensor Test with `sh pir-check pin_number`.
- **Unit:** any *PIN Address* (integer)
- **Example:** `-pin 4` (default)

#### 2. SET LENS TIME ADJUSTMENT

##### 2.1. `-lat time_amount:`

- Allows the user to select the Lens Adjusting Time for the camera to properly focus and regulate the light sensor.
- **Unit:** any *seconds* (integer)
- **Example:** `-lat 2` (default)

#### 3. SET MODEL CONFIDENCE

##### 3.1. `-c confidence_amount:`

- **Minimum confidence level required for the model to accept the result of the image elaboration process.** For example, *0.4* means *40%*.
- **Unit:** confidence [ *0.0 to 1.0* ] (float)
- **Example:** `-c 0.4` (default)

#### 4. SET RESOURCES PATH

##### 4.1. `-p /path/:`

- Allows the user to select another root path for the resources (labels, model, acquired image) to be loaded/saved.
- **Unit:** any *directory* (string)
- **Example:** `-p /home/pi/Documents/apps/pibird/` (default)

#### 5. SET DATASET SHIFT

##### 5.1. `-shift shift_amount:`

- **Allows the user to select an amount of shift in the labels file.** For example, the amount of shift is 2 if the labels list starts at the *2nd* place (index) in the file.
- **Unit:** any *index* (integer)
- **Example:** `-shift 2` (default)

#### 6. TOGGLE IG BOT

##### 6.1. `-ig toggle:`

- **Allows to enable/disable InstaBot and the InstaVault call.**
- **Unit:** toggle [ *1 or 0* ] (integer)
- **Example:** `-ig 1` (default)

**7. SET CORRECTION PROFILE**7.1. `-lprof profile_name:`

- **Allows the user to select a lens correction profile.** For now it either works with the *DDM-01* profile, or if a different profile is selected it will switch to *no-profile* mode.
- **Unit:** profile [ *none* | *DDM-01* ] *(string)*
- **Example:** `-lprof DDM-01` *(default)*

**8. SET PICTURE SIZE**8.1. `-size size_px:`

- **Allows to change the final image size.** This square (Ratio 1:1) image will be
- saved in the project directory and uploaded to Instagram. The model will be fed in parallel an image resized 224\*224 as per TF Model specifications.
- **Unit:** any *pixel* *(integer)*
- **Example:** `-ig 600` *(default)*

**9. PASS PERSONAL KEY (L2\_KEY)**9.1. `-l2 private_key:`

- **Allows to input the L2 Private Key required for the login with InstaVault In manual mode.**
- **Unit:** any *user L2 Private key* *(string)*
- **Example:** `-l2 myprivatekey`

**10. AUTORUN-AT-STARTUP**

- 10.1. Run `sudo nano /etc/xdg/lxsession/LXDE-pi/autostart` and add the line `@lxterminal --command="/home/pi/pibird"` if you wish to enable the autorun-at-startup functionality. Remove it if you wish to disable it.

## 4. COIN-DETECTOR

### 1. WHAT IS COIN-DETECTOR?

- 1.1. CoinDetector is an **AI Module** that allows you to detect and classify **2€ Coins** (soon 1€ Coins as well). Depending on parameters configuration, the AI will either completely **ignore** these kind of coins, or will **label** them as an **anomaly**:
- **Dirty, ruined or fake coins:** suited for application in which you have to **filter** these coins. Only back-facing coins are filtered.
  - **Rare coins, historical coins, mint errors:** suited for application in which you have to **spot** these coins for their commercial value. Only back-facing coins are spotted.
  - **Front-facing or other kind of coins:** suited for application in which you have to **spot** these coins. Usually hardly detected (even as anomaly).
  - **Common or commemorative coins:** suited for application in which you have to detect these coins. Only back-facing coins are detected. Up to 100% precision detection is performed.
  - **In general, common/recent commemorative coins are fully detected with a green color. While anomalies are partially detected with a light blue color.** The light blue squares on the anomalous coin are the pieces of the coin recognized by the AI, while the rest of the body of the coin has too many differences in respect to the originals thus resulting in a partial match.
- 1.2. The AI can accept both saved images and live webcam taken pictures (PiCoin, PiCoin Lite) as a source. The configuration can be suggested from the buyer at the moment of purchase. As a **good practice**, pictures containing **only images** should be provided to the AI. If any **text** is present in the picture, it could be labeled as an **anomaly**. If you want to filter it out, you can adjust the **maximum-bounding-boxes** parameter to the amount of actual coins in the picture (if known). Else, you could decrease the **minimum-score-threshold** parameter to exclude those weaker text activation while keeping the same amount of max-detections.

### 2. DISTRIBUTIONS

#### 2.1. LINUX DESKTOP

- **CoinDetector (GPU-ENABLED)**
  - **Connectivity:** OFFLINE
  - **Environment:** Linux Ubuntu (*Debian not yet tested*)
  - **CUDA:** 11.2
  - **cuDNN:** 8.1
  - **NVIDIA Driver:** 460/470
  - **Min Specs:**
    - **Intel i5** (required)
    - Execution speed depends on RAM, Disk and clock.
    - At least an **NVIDIA GTX 1070** is required to perform Hardware Acceleration.
- **CoinDetector Lite**
  - **Connectivity:** ONLINE
  - **Environment:** any
  - **Min Specs:**
    - **Intel i5** (required)
    - Execution speed depends on RAM, Disk and clock.



## 2.2. RASPBERRY-PI PORTABLE

- **PiCoin (GPU-ENABLED)**
  - **Connectivity:** OFFLINE
  - **Environment:** RaspbianOS
  - **Min Specs:** RaspberryPi 4, Model B
  - **Min GPU Specs:** NVIDIA JetSon
- **PiCoin Lite**
  - **Connectivity:** ONLINE
  - **Environment:** RaspbianOS
  - **Min Specs:** RaspberryPi 4, Model B.

## 3. USAGE

- **--path /path/:**
  - Allows to specify a full or relative path where the AI can find images to analyze.
  - **Unit:** any *directory* *(string)*
  - **Example:** `--path /usr/folder/`
- **-at anomaly\_threshold:**
  - Allows to specify a level below which partial entries are classified as anomalies.
  - **Unit:** threshold *[.0 to .99]* *(float)*
  - **Example:** `-at .975` *(default)*
- **-mbb max\_bounding\_boxes amount:**
  - Allows to specify the maximum amount of bounding boxes that can be printed on screen. It's best practice to set this number to the exact number of coins in the image (if known) or the approximate number of coins in the image in order to spot rare/fake coins with the greatest precision. Max 100 boxes are allowed in the same image.
  - **Unit:** amount *[1 to 100]* *(integer)*
  - **Example:** `-mbb 25` *(default)*
- **-mst minimum\_score\_threshold amount:**
  - Allows to specify the minimum % threshold for detections to be considered valid and so displayed on the screen, both as a 2EUR or ANOMALY recognition. If your picture contains text or foreign bodies, by tuning this parameter you might exclude such elements from the analysis.
  - **Unit:** threshold *[.0 to .99]* *(float)*
  - **Example:** `-mst .55` *(default)*

## 4. QUICKSTART

### 4.1. RUN COINDETECTOR OR PICOIN

- Run `sh coin-detector --path /your/path/here/` if you have the CoinDetector distribution to launch the application with the default parameters.
- Run `sh pi-coin --path /your/path/here/` if you have the PiCoin distribution to launch the application with the default parameters.
- This is the same as launching:  
`sh coin-detector --path /your/path/here/ -at .975 -mbb 25 -mst .25` because for every parameter you omit, the default values are used.

### 4.2. RUN WITH PARAMETERS

- Run `sh coin-detector` or `sh pi-coin` with one or more parameters at once, depending on the tuning you want to achieve.