ChatGPT

Object Detection for Penguins in Thermal Imagery with LiDAR Fusion

Introduction

Monitoring penguin colonies in natural environments (e.g. Antarctica and coastal areas) can be enhanced by **16-bit radiometric thermal imagery** combined with LiDAR and RGB data. Thermal cameras capture penguins as heat signatures, while LiDAR provides 3D structure (DSM/HAG) and RGB offers visual context. The goal is a detection system that outputs each penguin's location (bounding box or polygon with centroid), confidence score, and temperature statistics (mean, max, etc.), exportable to geospatial formats (GeoPackage/CSV). This report surveys suitable object detection models and methods – from deep learning detectors to multi-modal fusion techniques – capable of handling challenging cases: overlapping penguins, penguins partially obscured by bushes or inside burrows, eggs in burrows, and solitary penguins in varied poses (lying down, standing, moving). We prioritize approaches proven on **thermal/infrared imagery**, that can ingest **multi-modal (thermal + LiDAR + RGB)** data, and that are deployable in batch on desktop or cloud (no strict edge constraints).

Challenges in Penguin Detection (Thermal & LiDAR)

Detecting penguins in thermal imagery presents unique challenges. Key special cases include:

- **Overlapping individuals** – e.g. two penguins on top of each other may appear as one merged heat blob. The model must distinguish multiple peaks in one thermal region.
- **Partial occlusion by vegetation or terrain** – penguins at the edge or inside a bush, or occupying burrows in the ground, may be partly hidden in RGB imagery. Thermal can sometimes reveal a partial heat signature, and LiDAR can indicate the 3D shape of the bush or burrow opening.
- **Small objects (chicks or eggs)** – eggs in a burrow or very small chicks have a faint thermal signature and small size, requiring high-resolution detection and careful filtering to avoid false alarms from warm stones or noise.
- **Pose and background variability** – a lone penguin might be prone, standing, or moving. In thermal images, a lying penguin ("floating" on the ground) might have a different shape and heat distribution than a standing one. Background temperatures (snow, rock, soil, water) vary, and interestingly penguin feathers can sometimes be near ambient temperature, reducing contrast [1]. A robust system must handle varied thermal contrast scenarios.

16-bit radiometric data provides absolute temperature per pixel, enabling temperature-thresholding techniques. For example, setting a threshold (in °C) can isolate warm-bodied animals from a colder environment. However, absolute thresholds can be tricky outdoors – e.g. sun-warmed rocks or guano patches might register warm, and in some cases penguin exterior plumage can be as cold as ambient air [1]. Thus, models should be adaptable to use temperature information contextually rather than rely on a single cutoff. The high dynamic range of 16-bit imagery preserves subtle temperature differences that a model can exploit. Any deep learning model ingesting thermal frames should either accept calibrated float pixel values or normalized 16-bit values (to avoid losing the radiometric detail). The output requirements (polygons with centroid, confidence, and per-object temperature stats) mean the system should not only detect the object's location but also compute stats from the thermal values within that region. This is feasible since radiometric imagery allows extracting mean/max temperatures from the pixels inside a detection's mask or bounding box.

Detection Models for Thermal Imagery

One-Stage CNN Detectors (YOLO Family)

One-stage object detectors like **YOLO (You Only Look Once)** and its variants (v5, v7, v8, etc.) are popular for their speed and accuracy, and have been successfully applied to thermal aerial wildlife surveys [2] [3]. YOLO treats detection as a direct regression problem, predicting bounding boxes and class confidences in a single network pass [4]. Modern YOLO models incorporate Feature Pyramid Networks and anchor boxes, which help in detecting **small objects**. Notably, YOLOv7 introduced **focal loss** (to better handle class imbalance), which is *"ideal for identifying small objects"* [5]. In practice, researchers found YOLOv5 and YOLOv7 both effective for drone-based animal detection; YOLOv5 was slightly easier to train and in one study outperformed YOLOv7, possibly due to training stability and batch size limits [6] [3]. YOLOv8 (latest as of 2023) further improves accuracy but at higher computation cost.

Thermal-image training: YOLO models can be fine-tuned on thermal datasets by treating the single-channel infrared image as a pseudo-RGB (e.g. repeating the thermal channel into 3 channels or modifying the input layer to one channel). Pretraining on large visible datasets (like COCO) still provides useful feature representations. For example, in a NOAA study, CNN models (trained via the VIAME toolkit) achieved high accuracy on penguin detection in both RGB and IR imagery [7] [8]. Interestingly, that study found **no clear performance advantage of IR over color imagery** for detecting penguins given their data [9] – likely because images were taken in conditions where visible contrast was adequate. Nonetheless, YOLO models trained on thermal data have shown strong results in other cases: a recent study on multi-species detection found that fusing thermal with visible imagery greatly improved detection of cryptic animals (like deer in shadow) by **15–85%** in accuracy [10]. This suggests YOLO can leverage thermal cues effectively when the scenario demands it (e.g. low lighting or camouflaged animals). YOLO's fast inference makes it suitable for processing large mosaics or many video frames, though high-resolution thermal orthomosaics may need to be tiled into smaller sub-images to fit into GPU memory.

Strengths: YOLO is end-to-end trainable, easy to customize, and provides **bounding box + confidence** outputs directly. It can learn to implicitly combine temperature and shape cues from thermal data. The one-stage approach can recall small warm targets if appropriately tuned (e.g. using smaller anchor sizes and higher-resolution input). In one experiment on drone wildlife images, YOLOv5 achieved higher mAP for animal detection than a two-stage model, partly due to better training stability on the given dataset [6]. YOLO's architecture is also flexible enough to accept additional channels (for multi-modal input) with some modifications, though doing so requires careful weight initialization.

Limitations: Without explicit modifications, YOLO might struggle with very close-together penguins (it might output one box covering two penguins if they appear as one blob). It also has limited out-of-the-box ability to produce polygons or segmentations – it provides boxes, although one could approximate a polygon by the box or apply contour extraction on the heatmap inside the box. Overlapping penguins might require training the model to output multiple overlapping boxes, which one-stage detectors can do if the objects are distinguishable in features, but very closely packed penguins might still be merged. Post-processing (e.g. clustering or splitting predictions) might be needed in such cases.

Two-Stage and Segmentation Models (Faster R-CNN, Mask R-CNN)

Two-stage detectors like **Faster R-CNN** (region proposal CNN + classifier) and instance segmentation models like **Mask R-CNN** are also viable for thermal-based penguin detection. Faster R-CNN generates region proposals and can be more adept at separating adjacent objects, since the second stage examines each proposal in detail. **Mask R-CNN** extends this by predicting a segmentation mask per detected object, which is valuable for our case – the mask essentially gives a polygon outline of the penguin. Mask R-CNN has been used in the wild for thermal-based bird detection: for example, Kassim *et al.* (2020) employed Mask R-CNN to detect *"small faint objects"* (roosting turkeys) in nighttime drone IR videos [11] [12]. They used transfer learning (ImageNet-pretrained backbone) and achieved promising results in segmenting birds even against complex backgrounds [13]. The instance masks enabled them to **count overlapping birds** and distinguish them from tree branches or rocks. A post-processing step tracked detections across frames to eliminate false positives like stationary hot rocks – those spurious detections didn't persist over multiple frames, so they could be filtered out [11]. This highlights an advantage of segmentation + tracking: it can address false alarms by using temporal consistency. Even without video, segmentation models can incorporate shape constraints (e.g. the mask size/shape can be used to filter out non-penguin artifacts such as linear hot spots from rocks or equipment).

Strengths: Mask R-CNN can directly output a **polygon mask**, which matches our goal of polygonal outputs. The mask can also yield more precise temperature stats (mean/max over the penguin's area, excluding background pixels). Two-stage frameworks tend to have high localization accuracy and can separate touching objects if the algorithm learns to propose separate regions (the mask head can also split them). They may outperform one-stage detectors when background clutter or occlusions are present, due to the extra classification stage for proposals. Additionally, these models can incorporate **multi-modal input** by augmenting the backbone with extra channels or by feeding additional feature maps (e.g. a height map) into a second stream that merges with the main stream at some layer.

Limitations: Two-stage models are heavier in computation. Mask R-CNN in particular will be slower than YOLO – potentially an issue if thousands of 4K thermal frames or a huge mosaic need processing. However, for batch cloud or desktop processing, this may be acceptable. Training Mask R-CNN on limited thermal data might require careful augmentation because it has more parameters and could overfit. Also, annotation effort is higher (masks vs boxes). But if we have precise annotations (e.g. drawn polygons on thermal images), the payoff is a detector that can handle complex scenarios like penguins clustered together (the mask can conform to each body).

Traditional & Hybrid Approaches (Temperature Thresholding and Blob Detection)

In parallel to deep learning, **traditional image processing** can be surprisingly effective on thermal data, especially when penguins are significantly warmer or colder than their surroundings. A straightforward method is **absolute temperature thresholding**: pixels above a certain temperature (or within a range) are considered "hot spots" potentially corresponding to penguins. This yields a binary mask of candidate regions. Morphological operations (opening/closing) can refine this mask by removing noise and filling small gaps. Then, **connected component analysis** finds contiguous blobs, each of which can be a detection. This approach can quickly flag "something warm" in the scene. It's essentially how some simpler wildlife detection algorithms work, and it aligns with how a human might manually spot animals in a thermal image by eye.

The challenge is choosing the right threshold – thermal conditions vary. A **relative threshold** (e.g. pixels that are X °C above the local background average) can be more robust than a global cutoff. Some systems use a combination: first detect all warm regions liberally, then filter out those too small or with shapes not like a penguin. For instance, prior work counting birds in thermal images often did two-stage thresholding: one to find colony areas and another to isolate individuals [14]. In one approach for Adélie penguins, researchers manually isolated colony areas and then applied a brightness threshold to pick out individual penguin blobs [15]. This method was even packaged into an interactive MATLAB GUI for refining detections [16]. It demonstrates that thresholding + human correction can yield decent counts.

Modern workflows can hybridize this with AI. **Beta Informatics**, for example, implemented a LiDAR-based blob detection pipeline and a similar idea could apply to thermal. Their LiDAR pipeline creates a HAG (height-above-ground) raster and then masks cells within a certain height range (e.g. 0.2–0.6 m could correspond to penguin height) [17] [18]. They then do morphological opening/closing on this mask to remove noise, and label connected components [19]. Each connected region is a "penguin candidate." To handle multiple penguins clumped together, they optionally apply a **watershed segmentation**: for any large blob that likely contains multiple individuals, they identify local maxima (peaks) in the HAG surface and use those as seeds to split the blob [20] [21]. This successfully separates penguins standing close or one on top of another in the point cloud. A similar approach could be applied to thermal: large hot blobs can be split by finding peak heat points and using watershed on the temperature map.

After blob detection, Beta's pipeline filters out false positives by shape and context: regions with very elongated shapes or low solidity (likely vegetation or noise) are discarded [22], and any detections on steep terrain (where a rock outcrop might fool the height filter) are removed via a slope threshold [23]. The remaining candidates are output with properties like area, circularity, solidity, and the mean/max HAG value [24]. While this is done on LiDAR data, a similar feature-based filtering can refine thermal detections (e.g. filtering out extremely large warm areas like a sun-warmed rock shelf, which would have area/shape not consistent with a penguin). Temperature-specific criteria can help too, such as expected range of penguin body temperature (around 37–39 °C internally, though surface may be cooler). If an object's max temperature is only, say, 5 °C above background, it might not be a live penguin – or it could be a well-insulated one; thresholds must be tuned with caution.

Strengths: Traditional threshold + blob methods are **fast and interpretable**. They can leverage the **absolute temperature scale** (something generic CNNs won't inherently know) – for example, one could enforce that a detected blob's max temperature is within a realistic range for warm-blooded animals. These methods require no training data, which is useful if we have very few labeled examples initially. They also integrate well with GIS: the output is essentially a list of polygons or centroids that can be immediately saved to shapefiles or CSV. Beta Informatics' LiDAR tool, for instance, directly produces GeoJSON/GeoPackage of candidate points with attributes [25]. This is ideal for analysts who want to review and validate detections in QGIS or ArcGIS.

Limitations: Such methods can struggle with **false positives** (e.g. warm rocks, or sensor noise appearing as speckles) and **false negatives** if the threshold is too conservative. They also treat each warm blob as a penguin, which might not differentiate species or non-target warm objects (like seals, if present). There's no learned discrimination beyond temperature and size/shape, so performance might degrade in scenes where background objects have similar thermal signatures. A sensible approach is often to use thresholding as a **proposal generator**, then apply a CNN classifier to each proposal to confirm if it is a penguin (this could dramatically reduce the search space for the CNN, focusing it on likely areas).

In summary, a hybrid pipeline might be: threshold thermal image -> find blobs -> for each blob, extract features (size, mean temp, etc.) -> filter by simple rules -> optionally pass remaining regions through a small CNN to verify penguin vs false alarm. This could yield high precision and still leverage absolute temperature logic.

Multi-Modal Data Fusion (Thermal + LiDAR + RGB)

Integrating thermal imagery with LiDAR and RGB data can significantly enhance detection robustness. Each modality offers unique information:

- **Thermal IR:** Highlights warm-bodied animals against environment, even in darkness or shadow. As noted, it excels at detecting cryptic animals invisible in RGB (e.g. deer in shadow had up to 85% better detection with thermal fusion) [10]. However, thermal can be fooled by inanimate heat sources and may miss animals that are near ambient temperature.
- **LiDAR (DSM/HAG):** Captures the 3D structure of the terrain and objects. Penguins standing upright or moving on open ground are small ~0.5 m tall "bumps" on the ground – LiDAR can detect these bumps irrespective of temperature or color. A notable real-world example: a drone-mounted SICK 3D LiDAR could directly count penguins by their height/shape differences, even when their black feathers had low reflectance, reducing a multi-week ground count to a few hours of drone flight [26]. LiDAR can also identify features like burrows or bushes. A penguin *inside* a burrow might be invisible to LiDAR (as the laser won't penetrate the ground), but LiDAR can delineate the burrow's location and shape (depression in the DSM). Similarly, LiDAR can map bushes or rocks that might obscure penguins in RGB – knowing these occluders' geometry helps interpret a partial thermal hit.
- **RGB (Visual):** Provides high-resolution texture and color information. While penguins can be visually detected in RGB in clear conditions, their black/white contrast might actually be less useful from aerial view (they may appear as small dark dots, especially if resolution is low). RGB is most useful for context (e.g. differentiating penguins from seals or other species if color patterns differ, or identifying vegetation type). In our workflow, RGB might be mainly for annotation assistance or for a human to review the detections on a visible backdrop, rather than for the model, unless we specifically train a multi-channel model.

Fusion can occur at **data level, feature level, or decision level**:

- **Data level fusion (early fusion):** combining raw inputs into stacked channels. For instance, one could project the LiDAR-derived HAG or elevation map into the same 2D image grid as the thermal orthomosaic. The result might be a multi-layer image where one layer is temperature and another is height. A CNN could take these as a 2-channel image (or 3-channel with RGB as well). This early fusion gives the network the opportunity to learn cross-modal features (e.g. "warm blob with ~0.5 m height = penguin"). Some autonomous vehicle research follows this approach; e.g. Choi & Kim (2021) calibrated a thermal camera and LiDAR, mapping the LiDAR's depth into the thermal image, and fed both to a CNN to detect vehicles in low-visibility conditions [27] [28]. Early fusion requires precise alignment of data and careful normalization of each channel (temperatures and heights have different scales). In our case, producing a geo-accurate thermal **orthomosaic** aligned to the LiDAR

DSM is a critical preprocessing step. This is in progress in Beta's pipeline, using the LiDAR DSM to orthorectify each thermal frame [29]. Once aligned, each thermal pixel has a corresponding LiDAR elevation/height value and an RGB pixel (if an RGB ortho exists), enabling per-pixel fusion.

- **Feature level fusion (mid-fusion):** processing each modality with a separate model branch and then merging internal features. For example, one could have a ResNet that ingests the thermal image and another that ingests a DSM height map or an RGB image; the networks' feature maps might be concatenated at some stage (say after a few convolutional layers) and then fed to the detection head. This approach might capture modality-specific patterns better before mixing them. It's more complex to implement but has been explored in research (often in surveillance or self-driving domains, combining RGB and thermal for pedestrian detection). A new dataset "R-LiViT" (2025) even offers combined LiDAR, visual, and thermal data for such multi-sensor algorithms [30] [31].

- **Decision level fusion (late fusion):** running separate detectors on each modality and then combining their outputs. For instance, run a thermal-only detector to get thermal candidates, and run a LiDAR-based detector (like Beta's HAG blob detector) for LiDAR candidates. Then, cross-reference the results: a detection that appears in both thermal and LiDAR is very likely a true penguin (high confidence), whereas a thermal-only detection might be something like a warm rock (if LiDAR says no object protrusion there), and a LiDAR-only detection might be a penguin-shaped rock or moss (if thermal shows no heat, perhaps it's not an animal). Beta Informatics has explicitly planned such a fusion step: after getting LiDAR candidate centroids, they *"spatially join [each] LiDAR candidate to thermal pixels"* to extract the local thermal mean/max and a z-score of how hot that spot is relative to surroundings [32] [33]. Each candidate is then labeled as **LiDAR-only, Thermal-only, or Both** based on whether a significant thermal hotspot coincides with the LiDAR bump [33]. This can be used to triage results or to populate a confusion matrix if ground truth is known. Essentially, it's a logical AND/OR fusion: only flag as definite penguin if both modalities agree, or at least use the disagreement to prioritize verification. Decision fusion is very practical when one has distinct algorithms for each modality (which is our case – e.g. a CNN on thermal and a blob finder on LiDAR).

In general, **multi-modal fusion improves reliability**: Thermal may pick up a brooding penguin inside a burrow (which LiDAR misses as there is no height bump), and LiDAR may pick up a penguin-shaped statue (which thermal would correctly show as cold). By combining, such false positives/negatives can be resolved. There are indeed instances in ecology where one sensor alone was insufficient. Hinke *et al.* (2022) tested IR vs RGB for penguin and seal detection and found no clear advantage to IR alone [9] – but that doesn't mean IR was useless, just that RGB was also performing well in that scenario. On the other hand, Krishnan *et al.* (2023) found that fusing thermal+visible drastically helped in cases of *invisible-in-RGB* animals [10]. We can infer that in our project's polar context, if penguins are on snow or guano-covered ground, RGB contrast might be moderate, but if lighting is poor or penguins are dirty/burrowed, thermal will shine (figuratively and literally).

LiDAR & Thermal Fusion Tools: Phoenix LiDAR Systems – a leading provider of UAV LiDAR – supports multi-sensor payloads that can include thermal cameras. Their hardware kits (e.g. **Scout** series) allow mounting a FLIR thermal camera alongside the LiDAR, and their software suite ensures these streams are geo-aligned [34]. In fact, Phoenix notes that *"integrating RGB, thermal, or hyperspectral sensors with your LiDAR system has never been easier"*, enabling users to collect synchronized imagery and point clouds for **dynamic geospatial products** [34]. While Phoenix's proprietary software (SpatialExplorer, LiDARMill) focuses on LiDAR processing (e.g. point cloud generation, georeferencing, and visualization), one can use it to produce a **colorized point cloud** or **digital surface model**. For example, a thermal orthomosaic draped on a 3D point cloud could create a **radiometric 3D map**, where each point has a temperature value [35]. This is useful for manual analysis of penguin colonies in 3D or even automatic anomaly detection in point clouds. There is research on 3D thermal mapping via LiDAR SLAM and thermal cameras [36], indicating workflows to fuse these data for large-scale area scans. So, leveraging Phoenix's tools, one could obtain coregistered datasets: a radiometric orthomosaic TIFF and a precise DSM/HAG. From there, custom or open-source AI models would be applied; Phoenix doesn't provide animal detection algorithms out-of-the-box, but their ecosystem makes data prep easier. Beta Informatics, likely using such tools or equivalents, has an end-to-end pilot workflow: using a LiDAR-derived DSM to orthorectify thermal frames (via photogrammetry software and camera pose data), mosaicking them, and then performing LiDAR-thermal candidate fusion [29] [32]. The deliverable envisioned is a set of GeoTIFF tiles (thermal) and GeoPackage with detection points labeled whether they were seen in LiDAR, thermal, or both [33]. This kind of fused output would let an analyst query, for example, "give me all thermal-only detections" which might indicate potential misses in LiDAR (perhaps lying penguins or chicks), or "LiDAR-only" which might be false alarms like rocks shaped like penguins but not actually animals.

RGB fusion: We should note that incorporating RGB imagery, if available, could further help classification (distinguishing penguins from other warm-blooded animals like seals or birds in the same area). Multispectral data (as used by Bird *et al.* 2020) can identify guano stains to locate colony areas [37], effectively constraining the search area. In that study, a **semi-automated ArcGIS workflow** first used multispectral NIR imagery to detect red/brown guano patches as colony locations, then within those applied thermal to count individual penguins [37] [38]. The result was an automated count within ~4% of human counts on one island, and ~18% off on another (larger colonies had more deviation), with no statistically significant difference overall [38]. This underscores that **sensor fusion + targeted processing** (find colony, then individuals) can achieve high accuracy. In our context, if RGB or multispectral images are available, a similar strategy could be used: e.g. use RGB to mask out areas that obviously have no penguins (no guano, or areas of open water), or conversely use known colony locations as areas-of-interest for the thermal detector.

Models and Frameworks Comparison

To summarize the model options, the table below compares key attributes of representative approaches:

Model/ Method	Input Data	Output	Thermal Capability	Small/ Occluded Object Handling	Notes
YOLOv5/ YOLOv7 (One-stage CNN)	Thermal image (1-channel, treated as 3-channel or modified input). Optionally additional channels (if early fusion of LiDAR/ RGB).	Bounding boxes + confidence.	Proven on thermal with fine-tuning; one study used YOLOv5/7 on drone IR images and achieved high mAP [3] . IR alone gave similar performance to RGB in penguin detection in good conditions [9] . Can learn temperature features implicitly.	Multi-scale predictions and focal loss (YOLOv7) help with small objects [5] . May still merge overlapping warm blobs into one box without additional logic. Occlusions not explicitly addressed (needs good training examples).	Fast inference; large community support. Fine-tuning required on thermal data. Post-processing or tiling needed for very high-res images.

Model/ Method	Input Data	Output	Thermal Capability	Small/ Occluded Object Handling	Notes
Faster R-CNN / Mask R-CNN (Two-stage CNN)	Thermal image; can accept multi-channel (thermal + depth + RGB) via network modifications or separate branches.	Bounding boxes (Faster R-CNN) or boxes + segmentation mask (Mask R-CNN).	Successfully applied to thermal wildlife surveys (e.g. Mask R-CNN used for faint IR targets) [11] . Handles 16-bit data if normalized. Pretrained on COCO (RGB) then fine-tuned on thermal yields good results.	Better at separating touching objects due to proposal mechanism and mask segmentation. Mask R-CNN can produce distinct masks for overlapping penguins, and segment partial heat signatures. Still requires enough resolution to detect small objects; small proposal anchors need tuning.	Slower but more precise. Masks give polygon outlines for temperature stat extraction. Good for batch processing where accuracy > speed. Can integrate LiDAR by feeding additional input layers or processing LiDAR separately and merging results.

Model/ Method	Input Data	Output	Thermal Capability	Small/ Occluded Object Handling	Notes
Thresholding + Morphological Blob Detection (Traditional)	Thermal image (absolute temperature values). LiDAR HAG or DSM (for 3D blob detection).	Connected-component regions (can be converted to bounding boxes, centroids, or polygons).	Uses actual temperature thresholds (in °C) – e.g. identify pixels > 10 °C above background. Simple, no training needed. Needs tuning for environment; e.g. different threshold by time of day.	Natively handles small objects if threshold catches them. Overlapping objects appear as one blob – requires additional processing (e.g. watershed segmentation by finding multiple peaks) [21] . Occlusions: thermal can't see through objects, so hidden penguins not detected; LiDAR can detect shape even without heat.	Very fast. Useful as a pre-processing step or for candidate generation. High false-positive risk if environment has other warm objects (sunlit rocks, etc.). Beta's LiDAR pipeline is an example: it finds height-based "blobs" in point clouds within a size/height window [39] and splits/ clusters them to individual penguins [20] . Outputs are easily saved to GIS formats [25] .

Model/ Method	Input Data	Output	Thermal Capability	Small/ Occluded Object Handling	Notes
Multi-Modal Fusion Models (Custom)	Combined inputs (e.g. thermal as one channel, LiDAR height as another, RGB as third). Or separate CNNs for each modality.	Bounding boxes or masks (depends on base model).	Research prototypes exist (esp. RGB+Thermal for surveillance). No off-the-shelf model pre-trained for "thermal+LiDAR+RGB", so would require custom training. Potentially very powerful if training data covers the various scenarios.	By learning from multiple modalities, they can detect occluded penguins (e.g. thermal spike behind a bush + LiDAR shape of bush = penguin present). Small objects can be detected using complementary cues (a small thermal spot with a corresponding small elevation bump increases confidence).	Needs aligned and calibrated data. Higher complexity to train (must avoid one modality dominating learning). Could be achieved by extending YOLO or Mask R-CNN with extra input channels. Alternatively, implement late fusion: e.g. require agreement between thermal detector and LiDAR detector for high confidence detection [33].

Table: Comparison of detection model options for 16-bit thermal penguin imagery and multi-modal fusion.

Industry Solutions and Tools

Phoenix LiDAR Systems – Sensors & Software

While Phoenix LiDAR Systems primarily provides hardware (UAV LiDAR scanners and integrated sensor packages), their offerings are relevant to building a penguin detection system. Phoenix's drone mapping kits (e.g. **Scout-16, Scout-32, Ranger** series) allow mounting additional cameras including FLIR thermal sensors. The tight integration means the thermal imagery can be collected simultaneously with LiDAR, and with known calibration/offsets. Phoenix's spec sheets highlight that *RGB, thermal, or hyperspectral sensors can be added* to LiDAR systems to provide **"complex analysis"** and **"dynamic geospatial products"** for various applications [34]. In practice, a Phoenix system could produce: a georeferenced **point cloud (.LAS)**, an **orthomosaic thermal TIFF**, and possibly an **RGB orthomosaic**, all in the same coordinate frame. Their **SpatialExplorer** desktop software and **LiDARMill** cloud platform help process and align these data (SpatialExplorer can fuse RGB imagery to colorize the point cloud, which in principle could be done with thermal images as well by mapping temperature to color). This enables the creation of a 3D thermal model – essentially a point cloud where each point has a temperature value in addition to XYZ. Such a model allows slicing by height or extracting temperature of objects of certain size, which could be another way to detect penguins (e.g. find all point clusters of ~penguin size that have a high temperature median). Phoenix doesn't directly offer an AI detection algorithm for wildlife, but their tools would support the **data preprocessing and visualization** needed. A user could, for example, use Phoenix's output in CloudCompare or GIS software to manually validate detections: the LiDAR gives a "clean" .LAS of terrain and any standing objects, and the thermal mosaic draped on it would highlight which of those objects are warm (likely penguins).

Phoenix also often collaborates with partners for analytics; their hardware has been used in research like precision agriculture, infrastructure inspection, etc., where third-party AI models analyze the data. For penguin detection, a Phoenix system provides the *canvas* upon which the custom detection model operates. In summary, **Phoenix LiDAR's contribution** is in *data fusion hardware and software*: ensuring that collecting thermal + LiDAR data is as seamless as possible (important for scaling to large colonies), and providing post-processing pipelines (georeferencing, tiling, exporting) that our detection algorithms can plug into.

Beta Informatics – Custom Penguin Detection Workflow

Beta Informatics appears to be driving a bespoke solution for penguin detection using both LiDAR and thermal, tailored to ecological monitoring needs. From the available project notes and code snippets, Beta's approach is two-pronged:

- **LiDAR Route (Geometry-based detection):** They have a baseline module that takes LiDAR point clouds (LAS/LAZ from drone surveys) and converts them into gridded elevation models to find penguin-sized objects. This involves generating a **ground Digital Elevation Model (DEM)** and a **Height-Above-Ground (HAG) grid** by differencing the LiDAR returns with the ground surface [24] [40]. Then, they apply a height window (a min and max height) corresponding to expected penguin heights, and detect connected clusters within that window [17] [41]. Their implementation is efficient: it streams through the LiDAR data without loading all points at once [39], making it scalable to large areas. Initial pilot results showed this method successfully picks up "penguin-like detections" in the point clouds [42]. They further refine detections by removing those on steep slopes (assuming penguins don't stand on very steep terrain) and by checking shape descriptors (circularity, solidity) to ensure the point cluster looks like a compact blob rather than a long artifact [22]. Overlapping or clustered penguins in LiDAR are split by a watershed algorithm as described earlier [43] [44]. The LiDAR route outputs a set of candidate points each with attributes: e.g. X,Y coordinates, area of the blob, maximum and mean HAG (height) of the blob, etc. [24]. These can be written to GeoJSON/GPKG for mapping [25] or to CSV for further analysis. In essence, Beta's LiDAR module treats the 3D data like an image (which it is, after gridding) and applies classic blob detection tailored to penguin size/shape. This provides a list of *geometric detections* (could be penguin or any penguin-sized object).

- **Thermal Route (Radiometric detection):** In the current phase, Beta is focusing on producing a high-quality **thermal orthomosaic** that can be fused with the LiDAR data [45]. They plan to use the LiDAR-derived DSM to orthorectify oblique thermal images (since drone thermal cameras often have lower resolution and may be non-nadir) [29]. Once each frame is projected and stitched into a georeferenced mosaic (with tools like Correlator3D or Pix4D as hinted by their flowchart), they will have a temperature map over the colony area. On this thermal mosaic, a similar blob or AI detection can occur. The plan suggests computing per-candidate thermal statistics by sampling the mosaic at the LiDAR detection points [33]. Beta can label each LiDAR detection as:

- **Both** (thermal hotspot present at that location) – likely a true penguin,
- **LiDAR-only** (no thermal anomaly) – possibly a decoy (e.g. a rock or a deceased penguin), or an animal present but not radiating (unlikely if alive, but could happen if well insulated),
- **Thermal-only** (thermal shows a hot spot but LiDAR didn't flag an object) – could be a prone or hidden penguin (height too low to detect) or a warm object on the ground (e.g. sun-warmed rock).

This fusion will improve detection accuracy and provide a form of **confidence measure**. For example, a high-confidence penguin might be one detected by both methods. A thermal-only detection might be assigned lower confidence unless other evidence supports it. Temperature metrics like mean and max give additional insight – e.g. a thermal-only spot with a high max temperature (far above ambient) is more likely to be an animal (or something like an electronic device) than a mild warm patch. Beta's fusion output will include these metrics (they mention thermal mean/max and a normalized thermal z-score at each candidate) in the combined GeoPackage [33].

- **Other tools:** Beta's workflow references some third-party AI platforms (FlyPix, Picterra, etc. in the diagram). These could be off-the-shelf AI services where you can train a model on custom imagery. **Picterra**, for instance, is a cloud platform where users can upload orthomosaics and label a few examples (e.g. penguins) and it trains a detector in the cloud. It's known for geospatial object detection with an easy interface. Beta possibly evaluated these (Picterra, FlyPix) as alternative or complementary methods for detecting penguins on the orthoimagery. Such services can accelerate development: rather than coding a model from scratch, one might leverage them to get a quick detection model. However, they might not support multi-modal data directly (most accept just RGB or grayscale imagery). Still, if thermal is provided as a grayscale orthomosaic, a service like Picterra could be taught to detect heat signatures of penguins. The diagram also lists **Understory.ai** and **Treefera**, which sound like vegetation analytics platforms – perhaps considered for analyzing habitat (understory density, etc.) that could impact penguin detectability.

In summary, **Beta Informatics' solution** is shaping up to be a comprehensive system combining: LiDAR blob detection, thermal image processing, and data fusion in a GIS-friendly output. By custom-training or fine-tuning models on their data (should they choose to integrate a CNN like YOLO or Mask R-CNN), they can further improve on the initial blob detections. But even their current rule-based approach has shown promise in pilot tests. The modular design (Route A: LiDAR, Route B: Thermal, Route C: Multispectral in future) allows steady improvements and validation at each step [46] [47]. Notably, their emphasis on **batch processing and scalability** (streamed processing, Dockerized CLI, cloud mosaic generation) means the solution can be deployed on large survey datasets, not just frame-by-frame at the edge [39] [48]. This fits the requirement of desktop/cloud processing: heavy orthorectification and CNN computations can be run on a cloud instance or a powerful workstation after the drone data is collected, producing results for biologists to review.

Outputs and Integration

Achieving the desired outputs – polygons or boxes with centroids, confidence and temperature stats, in GeoPackage/CSV – is straightforward with the above methods. Deep learning detectors typically output bounding box coordinates (pixel or image coordinates). Once those are converted to map coordinates (using the orthomosaic's geotransform), centroids can be computed easily. For polygon outlines: if using Mask R-CNN, the predicted mask can be converted to a polygon (e.g. via marching squares algorithm) and then transformed to map coordinates. For blob detection methods, the connected component itself can be directly turned into a polygon (since it's essentially a binary mask shape). Beta's LiDAR code shows how each detection's image-row/col is mapped to real-world X,Y using the grid's cell size and origin [49] [50]. A similar transformation would apply for thermal image detections. Writing to **GeoPackage (GPKG)** can be done with GIS libraries (GDAL, Fiona, Geopandas in Python). CSV output can contain the centroid coordinates and attributes (temperature mean, etc.) for easy analysis in Excel or R.

The temperature statistics per detection can be computed by sampling the original 16-bit thermal orthomosaic at the pixels inside the detection region. For example, after YOLO detection one could take the bounding box, extract that sub-image and compute the mean and max temperature. If using segmentation, restrict to the mask area for stats (this avoids background pixels diluting the mean). "Contrast" could be defined as the difference between the object's temperature and the immediate surrounding background – one could compute the mean temperature in a ring around the detection and subtract, or use the z-score as Beta did (difference between object temp and global mean, divided by global standard deviation). These metrics help distinguish, say, a hot penguin vs a mildly warm patch.

Conclusion

Detecting penguins in 16-bit thermal imagery is a feasible task with modern computer vision models, and it is greatly enhanced by fusing complementary data like LiDAR and RGB. A prudent strategy is to **combine multiple methods** to harness their strengths: use deep learning (YOLO or Mask R-CNN) on the thermal images for adaptive, learned detection of penguin features, while also using rule-based detectors on LiDAR data to catch cases that thermal might miss. The two results can then be merged to improve overall accuracy and confidence. Multi-modal fusion ensures that even challenging cases (overlaps, occlusions, burrows) have at least one modality "seeing" the penguin. The surveyed research and pilot implementations back this up – e.g. dual-sensor drone surveys achieved higher detection rates [10] and automated counts rivaling manual ones [38].

In implementation terms, the system would involve: (1) **Data preprocessing**: generate georeferenced thermal mosaics and DSM/HAG from LiDAR (tools like Phoenix's software, Pix4D/Metashape, etc., assist here); (2) **Detection algorithms**: run a thermal detector (CNN or thresholding) and a LiDAR detector (height-based blob finder) in parallel; (3) **Fusion and filtering**: combine detections, assign confidences based on agreement and sensor-specific cues (temperature, height, shape); (4) **Output formatting**: save results as vector geospatial data with attributes for centroid, confidence, mean/max temp, etc. Each of these steps can be batch executed on a desktop or scaled on cloud infrastructure. None requires real-time processing, so models can afford to be deeper if needed.

By custom training models on the specific thermal signatures of penguins (and including difficult examples like two huddled penguins, penguin in bush, etc.), we can improve the detector's robustness to those scenarios. Data augmentation (simulating different backgrounds or partial occlusion) will help the CNN generalize. It's also wise to incorporate **negative samples** in training – e.g. warm objects that are not penguins – so the model learns to ignore those (the NOAA study found adding non-target annotations improved robustness a bit [51] [52]).

In conclusion, a hybrid approach using state-of-the-art object detectors alongside physics-based thresholding and multi-modal checks is recommended. This approach will produce reliable detection of penguins and even their eggs in complex outdoor scenes, with each detection enriched by temperature information. The combination of thermal and LiDAR data not only increases detection accuracy but also provides additional insights (like penguin height or burrow location) that can be valuable for ecological analyses. With tools from providers like Phoenix LiDAR Systems to gather and align the data, and the tailored algorithms from groups like Beta Informatics to analyze it, the goal of automated penguin detection and mapping can be achieved with high confidence.

Sources:

- Hinke *et al.*, 2022 – UAS thermal vs color imagery for penguin/seal detection [9] [7]
- Krishnan *et al.*, 2023 – Visible-thermal fusion improved animal detection (deer) [10]
- Phoenix LiDAR Systems – Multi-sensor LiDAR mapping solutions (thermal integration) [34]
- SICK AG – UAV LiDAR sensor counting penguins by 3D scanning (blog, 2017) [26]
- Bird *et al.*, 2020 – Multispectral (guano) + thermal fusion for Adélie penguin counts [37] [38]
- Kassim *et al.*, 2020 – Mask R-CNN for small bird detection in thermal videos [11]
- Beta Informatics internal pipeline notes – LiDAR blob detection and LiDAR-thermal fusion workflow [39] [33]
- Beta Informatics LiDAR code – morphological blob splitting, shape filtering, and geojson output for detections [43] [22] [50]
- Nature Scientific Reports 13:10385 (2023) – YOLOv5 vs YOLOv7 on fused thermal/RGB imagery [5] [3] .

---

[1] Penguins Can Be Colder Than Their Surroundings - FYFD
https://fyfluiddynamics.com/2013/04/thermal-imaging-of-emperor-penguins-in-antarctica/

[2] [3] [4] [5] [6] [10] Fusion of visible and thermal images improves automated detection and classification of animals for drone surveys | Scientific Reports
https://www.nature.com/articles/s41598-023-37295-7?error=cookies_not_supported&code=f2455b66-25f5-40c5-a47b-cec348a0bea9

[7] [8] [9] [51] [52] Evaluating Thermal and Color Sensors for Automating Detection of Penguins and Pinnipeds in Images Collected with an Unoccupied Aerial System
https://repository.library.noaa.gov/view/noaa/49307

[11] [12] [13] IS&T | Library
https://library.imaging.org/ei/articles/32/8/art00003

14 15 16 37 38 A Semi-Automated Method for Estimating Adélie Penguin Colony Abundance from a Fusion of Multispectral and Thermal Imagery Collected with Unoccupied Aircraft Systems
https://www.mdpi.com/2072-4292/12/22/3692

17 18 19 20 21 22 23 24 25 40 41 43 44 49 50 lidar_detect_penguins.py
file://file-EPb9UEtD7Jjj5yPsEYpSpq

26 3D LiDAR sensor detects the penguin even in its black tails | SICK
https://www.sick.com/ch/it/3d-lidar-sensor-detects-the-penguin-even-in-its-black-tails/w/blog-3d-lidar-sensor-detects-the-penguin-even-in-his-in-black-tails

27 28 manuscriptlink-society-file.s3-ap-northeast-1.amazonaws.com
https://manuscriptlink-society-file.s3-ap-northeast-1.amazonaws.com/kics/conference/icufn2021/abs/W-1B-2.pdf

29 32 33 39 42 45 46 47 48 CTA_meeting_081825.md
file://file-G46ZKLHtDrb6wnDAoUaX5v

30 31 [2503.17122] R-LiViT: A LiDAR-Visual-Thermal Dataset Enabling ...
https://arxiv.org/abs/2503.17122

34 phoenixlidar.com
https://www.phoenixlidar.com/wp-content/uploads/2020/06/PLS-SCOUT-16-Spec-Sheet_2020.pdf

35 3D Radiometric Mapping by Means of LiDAR SLAM and Thermal ...
https://pmc.ncbi.nlm.nih.gov/articles/PMC9653951/

36 3D Radiometric Mapping by Means of LiDAR SLAM and Thermal ...
https://www.mdpi.com/1424-8220/22/21/8512